异步图书
www.epubit.com

The Hundred-Page Machine Learning Book

机器学习精讲

[加拿大] 安德烈 • 布可夫（Andriy Burkov）著

韩江雷 译

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

□ □

ISBN：978-7-115-51853-8

[illegible]

[illegible] [[illegible]] [illegible]·[illegible]（Andriy Burkov）

[illegible] [illegible]

[illegible] [illegible]

[illegible] [illegible]11[illegible]

[illegible] 100164 [illegible] 315@ptpress.com.cn

[illegible] http://www.ptpress.com.cn

[illegible](010)81055410

[illegible](010)81055315

[illegible]

[illegible]

[illegible] 11 [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]5[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] “[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

2018[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

2019[illegible]

[illegible]

[illegible]

[illegible]——[illegible]“[illegible]”[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]·[illegible](Arther Samuel)[illegible]1959[illegible]IBM[illegible]IBM[illegible]2010[illegible]“[illegible]”(Cognitive Computing)[illegible]

[illegible]“[illegible]”[illegible]——[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]20[illegible]60[illegible]

[illegible]

[illegible]

[illegible]

[illegible]([illegible])[illegible]

[illegible]

[illegible]

第1章 [illegible]

1.1 [illegible]

[illegible],[illegible]
[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]
[illegible]

[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

1.2 [illegible]

[illegible]4[illegible]:[illegible](supervised)[illegible](semi-supervised)[illegible](unsupervised)[illegible](reinforcement)[illegible]

1.2.1 [illegible]

[illegible](supervised learning)[1][illegible](dataset),[illegible](labeled example)[2],[illegible]$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$[illegible]$N$[illegible],[illegible]$\boldsymbol{x}_i$[illegible](feature vector)[illegible]$j=1,\ldots,D$[illegible](feature),[illegible]$x^{(j)}$[illegible],[illegible]x[illegible],[illegible]

$x^{(1)}$[illegible],[illegible]$x^{(2)}$[illegible],[illegible]$x^{(3)}$[illegible],[illegible],[illegible]j[illegible]$\boldsymbol{x}_i$[illegible]$x_i^{(2)}$[illegible]([illegible]),[illegible],[illegible](label)y_i[illegible](class)[illegible]{1,2,…,C},[illegible],[illegible],[illegible],[illegible][3][illegible],[illegible],[illegible]——{“[illegible]”,“[illegible]”}[illegible]

[illegible](supervised learning algorithm)[illegible],[illegible],[illegible],[illegible]

1.2.2 [illegible]

[illegible],[illegible](unsupervised learning)[illegible],[illegible]$\{(\boldsymbol{x}_i)\}_{i=1}^N$[illegible](unsupervised learning algorithm)[illegible](model)[illegible]$\boldsymbol{x}$[illegible],[illegible]:[illegible](clustering)[illegible](cluster)[illegible];[illegible](dimensionality reduction)[illegible]$\boldsymbol{x}$[illegible];[illegible](outlier detection)[illegible],[illegible]$\boldsymbol{x}$[illegible]“[illegible]”[illegible]

1.2.3 [illegible]

[illegible](semi-supervised learning)[illegible],[illegible](semi-supervised learning algorithm)[illegible],[illegible],[illegible]

[illegible],[illegible]?[illegible],[illegible],[illegible]:[illegible],[illegible]

1.2.4 [illegible]

[illegible](reinforcement learning)[illegible],[illegible]“[illegible]”[illegible],[illegible](state)[illegible],[illegible],[illegible](policy)[illegible]

[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible]

这可能是机器学习最原始的场景了,尽管这个例子看似简单,但是对于许多复杂的机器学习问题都具有启发意义。为机器学习赋予计算能力最先是用于解决分类问题的,这类问题的核心在于定义合理的预测与分类的机器学习算法。

1.3 情感分析的一个简单例子

理解了分类问题,我们再来看看机器学习如何解决一个在现实中有意义的分类问题,在这里我们以情感分析为例。

情感分析是自然语言处理中的一项常见的分类任务(积极,消极)。给定一个句子或一段文本内容判断其情感倾向为积极还是消极,我们可以用人类的经验来判断一段文本中的情感倾向,例如,识别文本中出现的词汇(例如,"这部电影"、"真的很好看"、"好"、"坏"、"糟糕"等)来判断其情感,也可以通过更多特征(例如,文本中出现的感叹号是否超过4个以上)来判断[例如"好电影"中的若干个词("好的"、"伟大"、"精彩")]来判断其情感。

人类通常能够理解句子的意义并判断其中的情感,但机器没有这种能力,一台机器如何理解"这部电影"和"真的很好看"(句子中的词语,也就是机器学习的数据)的含义呢,这就需要将句子转化为机器能够理解的形式。

一种最简单的方式是将句子转化为数值向量(特征表示法),其中向量中的每个值对应于某个特征或某个词语是否出现在句子中。这种方法被称为"词袋模型"(bag of words)的表示方法,例如我们选取3 500个常用汉字作为字典[4](常用汉字表)中的词语来构造向量的维度。

- 第一维对应常用汉字表中的"一",如果句子中出现则赋值为1;否则为0。
- 第二维对应汉字表中的"乙",如果句子中出现则赋值为1;否则为0。
- 第三维对应汉字表中的"二",如果句子中出现则赋值为1;否则为0。

以此类推下去,则可以将一个句子转化为一个长度为3 500的向量表示法,其中的元素取值1或0。

□□,□□□□□□□□□□□□□,□□□,□□□□□0□1□□“□□□□□”□“□□□□”□□□□,□□□□□□□□□(Support Vector Machine,SVM)□□□□□□□□□□□□□□□□□□□□+1(□□1)□□□□□□(“□□□□”),□□□□□(“□□□□□”)□-1(□□1)□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□

SVM□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□,□□□□3 500□)□□□□□□□□□□□□□3 500□□□□□,□□□□3 499□□□□□(hyperplane)□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□(decision boundary)□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□$\boldsymbol{w}$,□□□□□□□$\boldsymbol{x}$□□□□□□□,□□□□□□b:

$$\boldsymbol{wx}-b=0$$

□□,$\boldsymbol{wx}$□□□□$w^{(1)}x^{(1)}+w^{(2)}x^{(2)}+\ldots+w^{(D)}x^{(D)}$,$D$□□□□□$\boldsymbol{x}$□□□□

□□□□□□□□□□□□□□□□□,□□□□□□2□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□2□□□□□□□□□

□□,□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{x}$□□□□:

$$y=\operatorname{sign}(\boldsymbol{wx}-b)$$

□□□sign□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□+1;□□□□□□,□□□-1□

SVM□□□□□□□□□□□□□□□□□**w**□*b*□□□□**w***□*b**□□□□□□,□□□□□(model)*f*(**x**)□□□□□□:

$$f(\boldsymbol{x})=\operatorname{sign}(\boldsymbol{w}^{*}\boldsymbol{x}-b^{*})$$

□□□,□□□,□**w***□□,□□□*b**□□□,sign□□□□□□□□□□□□(+1□□“□□□□”,-1□□“□□□□□”)□

□□,□□□□□□□□□**w***□*b**□?□□□□□□□□□□□(optimization)□□,□□□□□□□□

□□□□□□□□□?□□,□□□□□□□□□□□□□□□□□1□□□□□□□□□□□□□*i*=1,…,10 000□□□□□□($\boldsymbol{x}_i$,y_i),□□$\boldsymbol{x}_i$□□□□□□y_i□□□+1□-1□□□□□□□,□□□□□□□□□□:

$$\boldsymbol{w}\boldsymbol{x}_i-b\geqslant+1,\quad 当\ y_i=+1\ 时$$
$$\boldsymbol{w}\boldsymbol{x}_i-b\leqslant-1,\quad 当\ y_i=-1\ 时$$

□□□□,□□□□□□□□□□□□□□□□□□(margin)□□(generalization),□□□□□□□□□□□□□□□□□□□□,□□□□□□□**w**□□□□□□□(Euclidean norm)$\|\boldsymbol{w}\|$,□□□□□$\sqrt{\sum_{j=1}^{D}\left(w^{(j)}\right)^{2}}$□□□

□□□□,□□□□□□□□□□□□□□□□□□□□:

$$\operatorname{minimize}\|\boldsymbol{w}\|$$
$$\text{s. t. } y_i(\boldsymbol{w}\boldsymbol{x}_i-b)\geqslant 1,\quad 对于\ i=1,\cdots,N$$

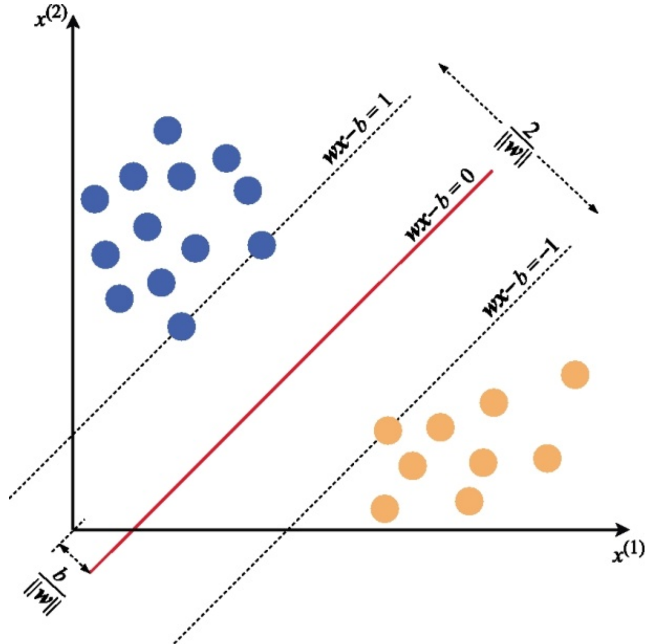
□□,□□□$y_i(\boldsymbol{wx}_i-b)\geq 1$□□□□□□□□□□

□□□□□□□$\boldsymbol{w}^*$□b^*□□□□□□(statistical model),□□□□□□□□□□□□□□

□□□□□□(training)□

□1.1□□□□□SVM□□□□□□□□1.1□,□□□□□□□□□□□,□□□□□□□□□□□□

□□□□;□□□□□□□□,□$\boldsymbol{wx}-b=0$□□□



□1.1 □□□□□□□□□□□□□□□SVM□□□□

□□,□□□□$\boldsymbol{w}$□□□□□□□□□□□□□□□□□□□□□□□□□□□?□□1.1□□,$\boldsymbol{wx}-b=1$□$\boldsymbol{wx}-b=-1$□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□$\frac{2}{\|\boldsymbol{w}\|}$□□□,□□$\|\boldsymbol{w}\|$□□,□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(linear model)□□□□□□□□,□□□□□□□□□□□□□□(□□□□□□□□□□□□□)□SVM□□□□□□□□□□

(kernel)[illegible],[illegible](outlier)([illegible]),[illegible]SVM[illegible][5][illegible]3[illegible]SVM[illegible]

[illegible],[illegible],[illegible],[illegible](accuracy,[illegible])[illegible]

[illegible],[illegible]:[illegible],[illegible],[illegible]

1.4 [illegible]

[illegible]?[illegible]1.1[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],SVM[illegible],[illegible]

[illegible],[illegible],[illegible](learnability)[illegible]PAC(Probably Approximately Correct)[illegible]PAC[illegible]

---

[1] [illegible]

[2] [illegible]:“[illegible]”[illegible]“[illegible]”“[illegible]”[illegible],[illegible]

[3] [illegible],[illegible]0[illegible]256.34[illegible]1000[illegible]1000.2[illegible]

[4] [illegible]:[illegible]20000[illegible]

[5] [illegible],[illegible]([illegible])[illegible],[illegible]

[illegible]2[illegible] [illegible]

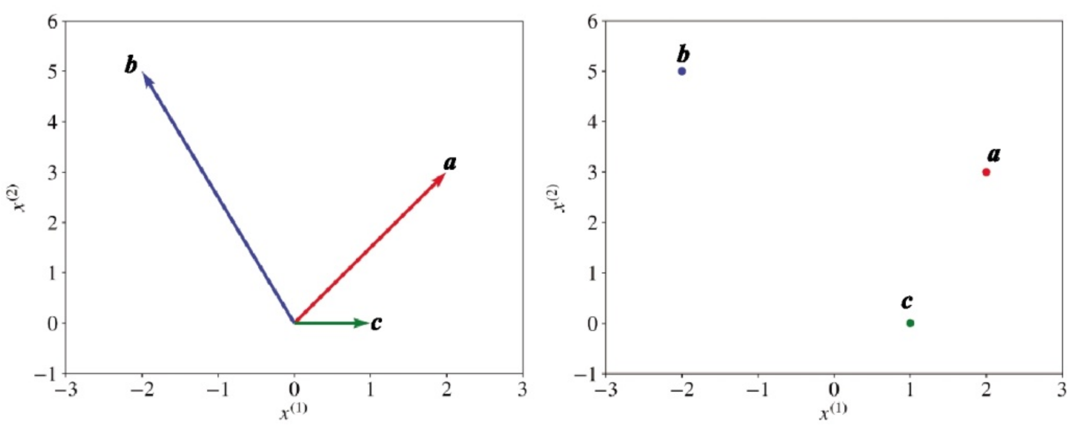
2.1 [illegible]

[illegible]

2.1.1 [illegible]

[illegible](scalar)[illegible],[illegible]15[illegible]-3.25[illegible],[illegible]x[illegible]a[illegible]

[illegible](vector)[illegible](ordered list)[illegible],[illegible],[illegible]$\boldsymbol{x}$[illegible]$\boldsymbol{w}$[illegible],[illegible],[illegible]2.1[illegible]3[illegible],[illegible]$\boldsymbol{a}$=[2,3][illegible]$\boldsymbol{b}$=[-2,5][illegible]$\boldsymbol{c}$=[1,0][illegible],[illegible]$w^{(i)}$[illegible]$x^{(j)}$[illegible],[illegible]i[illegible]j[illegible],[illegible]2.1[illegible]$\boldsymbol{a}$[illegible],$\boldsymbol{a}^{(1)}$[illegible]$\boldsymbol{a}^{(2)}$[illegible]2[illegible]3[illegible]



□2.1　3□□□□□□□□3□□□□3□□

□□□□□□,$x^{(j)}$□□□□□,□□x^2(□□)□x^3(□□)□□□□□□□□□□□□□□□□□□□□□□□□□,□□□($x^{(j)}$)^2□□□□

□□□□□□□□□□□□□□,□□$x_i^{(j)}$□$x_{i,j}^{(k)}$□□□□□□□,□□□$x_{l,u}^{(j)}$□□□l□□□□□□u□□□□□j□□□□□□

□□□□(matrix)□□□□□□□□□□□□□□□□(array)□□□□□□□□□□□□□□□□□

$$\begin{bmatrix} 2 & 4 & -3 \\ 21 & -6 & -1 \end{bmatrix}$$

□□□□□□□□□□□□□,□□$\boldsymbol{A}$□$\boldsymbol{W}$□

□□□□(set)□□□□□□□□□□□□□□□(unordered collection)□□□□□□□□□□□□(□S)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□{1,3,18,23,235}□{$x_1,x_2,x_3,x_4,\dots,x_n$}□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□a□b□□□□□□,□□a□b□□,□□□□□□□□□□□□□□[a,b];□□a□b□□□□□□,□□□□□□□□□□(a,b)□□□□□,□□[0,1]□□□□□□□□:0□0.000 1□0.25□0.784□0.999 5□1.0□□□,□□□□□$\mathbb{R}$□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

$x \in S$□□“x□□□□□□S”□□□□□S_1□S_2□□□(intersection)□□□□□□□S_3,□□□$S_3 \leftarrow S_1 \cap S_2$□□□,{1,8}←{1,3,5,8}∩{1,8,4}□

S_1□S_2□□□(union)□□□□□□□□S_3,□□□$S_3 \leftarrow S_1 \cup S_2$□□□,$\{1,3,4,5,8\} \leftarrow \{1,3,5,8\} \cup \{1,8,4\}$□

2.1.2 □□□□□(∑)□□

□□□□□$\boldsymbol{X}=\{x_1,x_2,\ldots,x_{n-1},x_n\}$,□□□□□$\boldsymbol{x}=[x^{(1)},x^{(2)},\ldots,x^{(m-1)},x^{(m)}]$□□□□□□□□□□,□□□∑□□:

$$\sum_{i=1}^{n} x_i \stackrel{\text{def}}{=} x_1 + x_2 + \ldots + x_{n-1} + x_n$$

□□

$$\sum_{j=1}^{m} x^{(j)} \stackrel{\text{def}}{=} x^{(1)} + x^{(2)} + \ldots + x^{(m-1)} + x^{(m)}$$

□□,□□$\stackrel{\text{def}}{=}$□□□□“□□□”(is defined as)□

2.1.3 □□□(Π)□□

□□□□□(Π)□□□□□□□∑□□□Π□□□□□□□□□□□□□□□□□□:

$$\prod_{i=1}^{n} x_i \stackrel{\text{def}}{=} x_1 \cdot x_2 \cdot \ldots \cdot x_{n-1} \cdot x_n$$

□□,$a \cdot b$□□a□□b□□□□,□□□□□□□□□ab□□□a□□b□

2.1.4 □□□□

使用集合构造符号还可以对集合进行某些变换:$S' \leftarrow \{x^2 | x \in S, x > 3\}$。上面这个表达式中,新集合$S$只包括大于3的元素的平方值,而且条件运算符确保$S'$的$|S|$可以小于或等于$S$的元素数量。

2.1.5 向量运算

向量的加法:$\boldsymbol{x}+\boldsymbol{z}=[x^{(1)}+z^{(1)},x^{(2)}+z^{(2)},\ldots,x^{(m)}+z^{(m)}]$

向量的减法:$\boldsymbol{x}-\boldsymbol{z}=[x^{(1)}-z^{(1)},x^{(2)}-z^{(2)},\ldots,x^{(m)}-z^{(m)}]$

向量与标量相乘将产生一个向量,例如:

$$\boldsymbol{x}c \stackrel{\text{def}}{=} \left[cx^{(1)}, cx^{(2)}, \ldots, cx^{(m)}\right]$$

两个向量的点积(dot product)或内积(inner product)是一个标量,例如

$\boldsymbol{w}\boldsymbol{x} \stackrel{\text{def}}{=} \sum_{i=1}^{m} w^{(i)} x^{(i)}$ 。在一些书里,点积以$\boldsymbol{w}\cdot\boldsymbol{x}$表示。两个向量必须同维,否则点积是未定义的,作为向量运算不成立。

一个矩阵$\boldsymbol{W}$和一个向量$\boldsymbol{x}$相乘会产生另一个向量,设矩阵示例如下:

$$\boldsymbol{W} = \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix}$$

当矩阵和向量参与运算的时候,向量以一个具有一列多行的矩阵表示。只有当向量和矩阵的行数相同时,矩阵和向量才能相乘。设向量为$\boldsymbol{x} \stackrel{\text{def}}{=} [x^{(1)}, x^{(2)}, x^{(3)}]$,则$\boldsymbol{W}\boldsymbol{x}$是一个二维向量,其具体运算方式如下:

$$
\boldsymbol{W}\boldsymbol{x} = \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}
\stackrel{\text{def}}{=} \begin{bmatrix} w^{(1,1)}x^{(1)} + w^{(1,2)}x^{(2)} + w^{(1,3)}x^{(3)} \\ w^{(2,1)}x^{(2)} + w^{(2,2)}x^{(2)} + w^{(2,3)}x^{(3)} \end{bmatrix}
= \begin{bmatrix} \boldsymbol{w}^{(1)}\boldsymbol{x} \\ \boldsymbol{w}^{(2)}\boldsymbol{x} \end{bmatrix}
$$

□□□□□□□□□□□□□□□□,□□***W***□□□,□***x***□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□□□,□□□□□(transpose)□□□□□□□□□□□□□ ***x***$^{\mathrm{T}}$□□***x***□□□□□□□□:

$$
x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}, \quad □ x^{\mathrm{T}} \stackrel{\text{def}}{=} \left[x^{(1)}, x^{(2)}\right]
$$

□□***x***□□□***W***□□□□□□□***x***$^{\mathrm{T}}$***W***□□□□□□□:

$$
\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W} = \left[x^{(1)}, x^{(2)}\right] \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix}
\stackrel{\text{def}}{=} \left[w^{(1,1)}x^{(1)} + w^{(2,1)}x^{(2)}, w^{(1,2)}x^{(1)} + w^{(2,2)}x^{(2)}, w^{(1,3)}x^{(1)} + w^{(2,3)}x^{(2)}\right]
$$

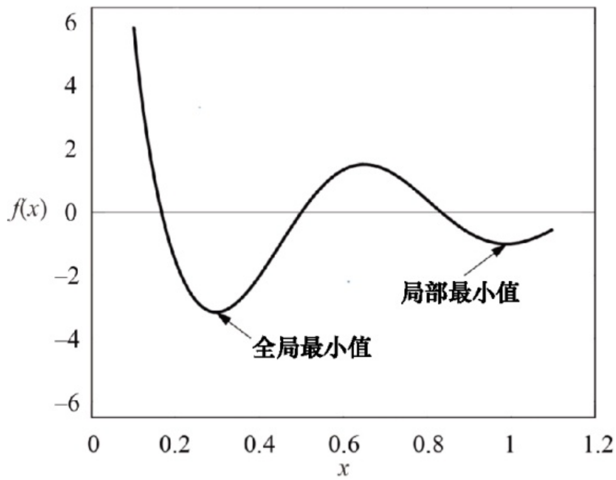
□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□

2.1.6 □□

□□□□(function)□□□□□□□□*X*□□□□*x*,□□□□□□*Y*□□□□*y*□□□□□□□□□ *X*□□□□□□□□(domain),*Y*□□□(co-domain)□□□□□□□□□□□□□□□□,□□□ □□□□□□□*f*,□□□□□□□□*y*=*f*(*x*),□□“*y*□*x*□□□”□□□,*x*□□□□(argument),□ □□□;*y*□□□□,□□□□□

□□□□□□□□□□□□□□□□(variable)□□□:x□□□f□□□□

□□□□□□$x=c$□□□□□□□□,□□x□□□$f(x)\geq f(c)$,□□□$x=c$□$f(x)$□□□□□□□(local minimum)□□□□□(interval)□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□□□□□□□□□(open interval)□□□□□□□□□□□□□□□,□□□□□□□□□□:(0,1)□□□□□□□□□□□□0□□□1□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□2.2□□□



□2.2 □□□□□□□□□□□□□□

$\boldsymbol{y}=f(x)$□□□□□□□□□□□□□□$\boldsymbol{y}$□□□□□□□□□□□□□□□□□□□□□□□

2.1.7 max□argmax

□□□□□□$A=\{a_1,a_2,\ldots,a_n\}$,□□□$\max\limits_{a\in A} f(a)$□□□□:□$A$□□□□□□□□,$f(a)$□□□□□□□□□$\operatorname{argmax}\limits_{a\in A} f(a)$□□□□$f(a)$□□□□□$a$□

□□□□□□□□□□□□□□□□□□□,□□□□$\max\limits_{a} f(a)$□$\operatorname{argmax}\limits_{a} f(a)$□

□□□min□argmin□□□□max□argmax□□□

2.1.8 □□□□□

□□□$a \leftarrow f(x)$□□□□,□□a□□□□□□□$f(x)$□□□□a□□□□□□□□□□,$\boldsymbol{a} \leftarrow [a_1, a_2]$□□□□□□□□□□□□□□□□□□□$[a_1, a_2]$□

2.1.9 □□□□□

□□□□□□□(derivative)f'□□□□□□□□□□f□□(□□□)□□□□□□□□□□□□□□,□□5□-3,□□□□□f□□□□□□□□x□□□(□□□)□□□□□□□□□□□□f'□□□□□,f□□□□□□□□□□□□□□□□□□□□□f'□□□x□□□□,f□□□□□;□□□□□□,□f□□□□□□□□□□□x□□0,□□□□□□□x□□□□(slope)□□□□□

□□□□□□□□□□□(differentiation)□

□□□□□□□□□□□□□□□□,□□$f(x)=x^2$,□$f'(x)=2x$;□□$f(x)=2x$,□$f'(x)=2$;□□$f(x)=2$,□$f'(x)=0$(□□□□□□$f(x)=c$□□□□□0)□

□□□□□□□□□□(chain rule)□□□□□□□□□□□□□□□,□$F(x)=f(g(x))$,f□g□□□□,□$F'(x)=f'(g(x))g'(x)$□□□□□,□□$F(x)=(5x+1)^2$,□□$g(x)=5x+1$□$f(g(x))=(g(x))^2$□□□□□□□,□□□□$F'(x)=2(5x+1)g'(x)=2\times(5x+1)\times5=50x+10$□

□□□□□□□□□□□□□□(□□□□□□□□□□,□□□□),□□□□□□□□□(generalization)□□□(gradient)□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□

□□,□□□□□□□$f([x^{(1)},x^{(2)}])=ax^{(1)}+bx^{(2)}+c$,$\frac{\partial f}{\partial x^{(1)}}$□□□$x^{(1)}$□□□$f$□□□□,□:

$$\frac{\partial f}{\partial x^{(1)}} = a + 0 + 0 = a$$

□□,a□□□$ax^{(1)}$□□□;□□0□□□$bx^{(2)}$□c□□□□□□□□□□$x^{(1)}$□□□□,□□$x^{(2)}$□□□□□□□,□□$bx^{(2)}$□□□□0□

□□,□□□$\frac{\partial f}{\partial x^{(2)}}$□□□$x^{(2)}$□□□$f$□□□□,□:

$$\frac{\partial f}{\partial x^{(2)}} = 0 + b + 0 = b$$

□□,□□f□□□,□□□∇f,□□□□□$\left[\frac{\partial f}{\partial x^{(1)}}, \frac{\partial f}{\partial x^{(2)}}\right]$□

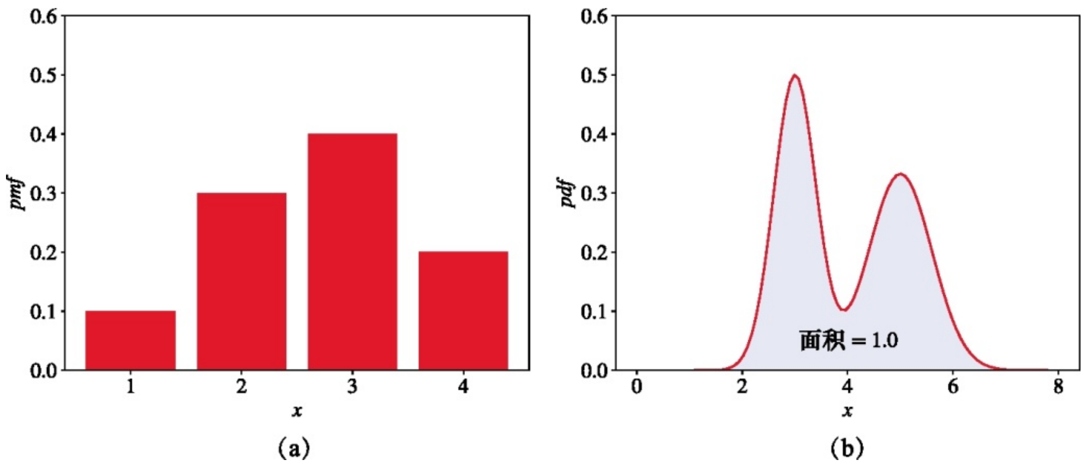
□□4□□,□□□□□□□□□□□□□□□□□□□□

2.2 □□□□

□□□□□□(random variable)□□□□□□□(random phenomenon)□□□□□,□□□□□□□□□□□□(□X)□□□□□□□□□□:□□□(discrete)□□□□(continuous)□

□□□□□□□□□(discrete random variable)□□□□□□□□□□□□□□□□□□□□,□□,□□,□□□□1,2,3□□

□□□□□□□□□□□□□□□□□□□□□□□(probability distribution),□□□□□□□□□□□□□□,□□□□□□□□□(probability mass function,pmf)□□□,Pr(X=□□)=0.3,Pr(X=□□)=0.45,Pr(X=□□)=0.25□□□□□□□□□□□□□□□□□□□0,□□□□□□□□□□□1[□□2.3(a)]□



□2.3 □□□□□□□□□□□□□□□□□

□□□□□□□□□(continuous random variable)□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□X□□□□□□□□□,□□□□□□□c□□□Pr(X=c)□□0□□□□□□□□□□□□(probability density function,pdf)□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□1[□□2.3(b)]□

□□□□□□□□□□□□□X□k□□□□,□□□$\{x_i\}_{i=1}^{k}$□□□□□X□□□(expectation)$\mathbb{E}[X]$□:

$$\mathbb{E}[X] \overset{\text{def}}{=} \sum_{i=1}^{k} \boldsymbol{x}_i \cdot \Pr(X = x_i) = x_1 \cdot \Pr(X = x_1) + x_2 \cdot \Pr(X = x_2) + \ldots + x_k \cdot \Pr(X = x_k) \tag{2.1}$$

□□,Pr(X=$\boldsymbol{x}_i$)□□□□pmf,X□□x_i□□□□□□□□□□□□□□□□□□□□(mean,average),□□□□□□□μ□□□□□□□□□□□□□□□□□□□(statistics)□□□

□□□□□□□□□□□□□□□(standard deviation),□□□:

$$\sigma \stackrel{\text{def}}{=} \sqrt{\mathbb{E}\left[(X-\mu)^2\right]}$$

□□(variance),□□□σ^2□var(X),□□□:

$$\sigma^2 = \mathbb{E}\left[(X-\mu)^2\right]$$

□□□□□□□□□□□□□□□□□□□□:

$$\sigma = \sqrt{\Pr\left(X=x_1\right)\left(x_1-\mu\right)^2 + \Pr\left(X=x_2\right)\left(x_2-\mu\right)^2 + \ldots + \Pr\left(X=x_k\right)\left(x_k-\mu\right)^2} \tag{2.2}$$

□□,$\mu = \mathbb{E}[X]$□

□□□□□□□□□□□□X□□□□:

$$\mathbb{E}\left[X\right] \stackrel{\text{def}}{=} \int_{\mathbb{R}} x f_X\left(x\right) \mathrm{d}x \tag{2.3}$$

□□,f_X□□□X□pdf,$\int_{\mathbb{R}}$□□□□xf_X□□□(integral)□□□

□□□□□□□□□□□□□,□□1,□□□□□□□□□□$\int_{\mathbb{R}} f_X(x)\mathrm{d}x=1$□

[illegible],f_X[illegible],[illegible]X[illegible],[illegible](example)[illegible](sample)[illegible](dataset)[illegible]

2.3 [illegible]

[illegible],[illegible]$S_X = \{x_i\}_{i=1}^{N}$[illegible]f_X[illegible],[illegible],[illegible](unbiased estimator)[illegible]

[illegible]S_X,[illegible]θ[illegible],[illegible]$\hat{\theta}(S_X)$[illegible]

$$\mathbb{E}\left[\hat{\theta}\left(S_X\right)\right] = \theta$$

[illegible],$\hat{\theta}$[illegible](sample statistic)[illegible],[illegible]$\hat{\theta}$[illegible]S_X[illegible],[illegible]θ[illegible]X[illegible]X[illegible],[illegible],[illegible],$\hat{\mu}$,[illegible]X[illegible]μ[illegible]$\hat{\mu}$[illegible]

[illegible]2.1[illegible]2.3[illegible],[illegible]$\mathbb{E}[X]$[illegible]$\frac{1}{N}\sum_{i=1}^{N} x_i$[illegible][[illegible](sample mean)][illegible]

2.4 [illegible]

[illegible](conditional probability)[illegible]Y[illegible]y[illegible],[illegible]X[illegible]x[illegible],[illegible]$\Pr(X=x|Y=y)$[illegible](Bayes'Rule)[illegible](Bayes'Theorem),[illegible]:

$$\Pr(X=x|Y=y)=\frac{\Pr(Y=y|X=x)\Pr(X=x)}{\Pr(Y=y)}$$

2.5 [illegible]

[illegible]X[illegible],[illegible]θ[illegible]f_θ[illegible],[illegible]μ[illegible]σ[illegible](Gaussian function)[illegible]

$$f_\theta(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

[illegible],$\theta\overset{\text{def}}{=}[\mu,\sigma]$[illegible]

[illegible],[illegible]X[illegible],[illegible]θ[illegible]:

$$\Pr(\theta=\hat{\theta}|X=x)\leftarrow\frac{\Pr(X=x|\theta=\hat{\theta})\Pr(\theta=\hat{\theta})}{\Pr(X=x)}$$

$$=\frac{\Pr(X=x|\theta=\hat{\theta})\Pr(\theta=\hat{\theta})}{\sum_{\tilde{\theta}}\Pr(X=x|\theta=\tilde{\theta})\Pr(\theta=\tilde{\theta})}$$

[illegible],$\Pr(X=x|\theta=\hat{\theta})\overset{\text{def}}{=}f_{\hat{\theta}}$[illegible]

[illegible]X[illegible]S,[illegible]θ[illegible],[illegible],[illegible]$\Pr(\theta=\hat{\theta})$[illegible]$s\in S$[illegible]$\Pr(\theta=\hat{\theta})$[illegible]$\sum_{\hat{\theta}}\Pr(\theta=\hat{\theta})=1$[illegible]$\hat{\theta}$[illegible](prior)[illegible]

[illegible],[illegible]$\hat{\theta}$[illegible]$\Pr(\theta=\hat{\theta}|X=x_1)$[illegible],[illegible]2.4[illegible]$x=x_2\in S$[illegible]$\Pr(\theta=\hat{\theta}|X=x)$[illegible],[illegible],[illegible]

$$\Pr(\theta = \hat{\theta}) \leftarrow \frac{1}{N} \sum_{x \in S} \Pr(\theta = \hat{\theta} | X = x)$$

□□□□□□□$\Pr(\theta = \hat{\theta})$□

□□□□□□,□□□□□□□□□□□(maximum likelihood)□□□□□□□□□□θ^*:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{N} \Pr\left(\theta = \hat{\theta}\, |X| = x_i\right)$$

(2.5)

□□θ□□□□□□□,□□□□□□□□□□□□□[□□□□□(gradient descent)□]□□□□□□2.5□□□□□□□□□□4□□□□□□□□□□□,□□□□□□□2.5□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□[1]□

2.6 □□□□□

□□□□□□□□□□□□□□□,□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□5□□□□□□□□□□

□□□□□□□□□,□□

2.7 □□vs.□□

□□(classification)□□□□□□□□□□□(unlabeled example)□□□□(label)□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□□(classification algorithm)□□□□□□□□□□□□□□□□(labeled example)□□□□,□□□□□□(model)□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□(□□□□□□)□

□□□□□□□□□□□□□□□□□□(classe)□□□□□□□□□□□□□□(“□□”/“□□”,“□□”/“□□□”),□□□□□□□□□□□□(binary classification)[□□□□□□□□□□□□(binomial)]□□□□□□□□□□3□□□□□□□□[2],□□□□□□□□□□□□(multiclass classification)□

□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□7□□□□

□□(regression)□□□□□□□□□□□□□□□□,□□□□□□□□□□(□□□□□)□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□(regression learning algorithm)□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□

2.8 □□□□□□vs.□□□□□□

□□□□□□□□□□□□□□□□□□□,□□□□□□1□□□□□SVM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□SVM□□,□□□□□□□□□□□□$\boldsymbol{w}^*$□b^*□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□**k**□□(k-Nearest Neighbor,kNN)□□□□□□□□□□□□,kNN□□□□□□□□□□*k*□□□□□□□□□□□□□“□□”□□□□□□□□□□,kNN□□□□□□□□□□□□

2.9 [illegible]vs.[illegible]

[illegible](shallow learning)[illegible](neural network)[illegible],[illegible](layer)[illegible],[illegible](deep neural networks)[illegible](“[illegible]”)[illegible],[illegible],[illegible]

[illegible]6[illegible]

---

[1] [illegible],[illegible],[illegible]

[2] [illegible]

□3□　□□□□

□□□□,□□□□□5□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□

3.1　□□□□

□□□□(linear regression)□□□□□□□□□,□□□□□□□□(linear combination)□□□□□

3.1.1　□□□□

□□□□□□□□□$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$,$N$□□□□□,$\boldsymbol{x}_i$□□□□$i$=1,…,$N$□$D$□□□□□$y_i$□□□□□□[1],□□□$x_i^{(j)}$,$j$=1,…,$D$□□□□□□

□□□□□□□□□□□□x□□□□□□□$f_{\boldsymbol{w},b}(\boldsymbol{x})$:

$$f_{\boldsymbol{w},b}(x) = \boldsymbol{w}\boldsymbol{x} + b \tag{3.1}$$

$f_{\boldsymbol{w},b}$□□f□□□□□□□□,□$\boldsymbol{w}$□b□□□□□□,$\boldsymbol{w}$□□□□□□D□□□,b□□□□□

□□□□,□□□□□□□□□y□□$\boldsymbol{x}$□□□□,□□$y \leftarrow f_{\boldsymbol{w},b}(\boldsymbol{x})$□□□□□□□□,□□□□□□□□($\boldsymbol{w}$,$b$)□□□□□□□□□□□□□□□□□□□□□□□□□□($\boldsymbol{w}^*$,b^*)□□

变量的类别,两者都是分类算法。

对于回归问题来说,以图3.1为例,与线性分类、SVM等方法不同的是,回归不需要sign函数的转化,而是直接进行输出的,这就造成了与SVM等分类算法的本质不同,回归模型的输出是连续变量,而非分类算法中的离散的类别变量。

在线性回归中,我们通过拟合一条直线来解释输入输出之间的关系,如图3.1所示。线性回归模型的目的便是找到一个线性函数(红线)以最大程度拟合数据(黑点)。当有了新数据,即输入一个新的输入x_{new}来预测y_{new}。当输入数据的维度D为$(D>1)$,回归模型拟合的模型不是一条直线(这里是二维的)而是一个超平面$(D>2)$。

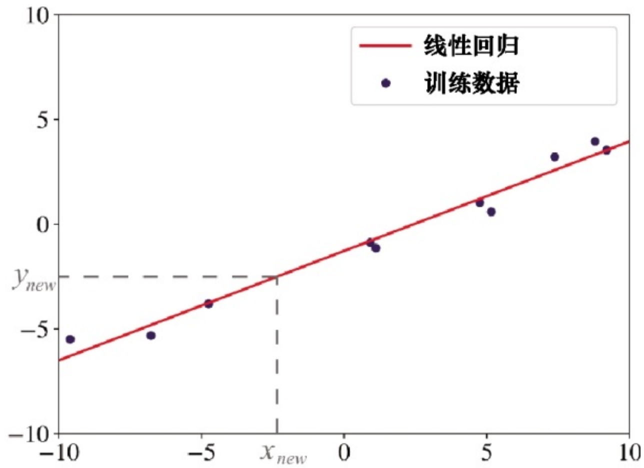


图3.1　10个点的线性回归示意图

我们希望拟合的这个模型能够对新的数据具有好的预测能力:比如图3.1中给定一个新的训练数据上的点,我们能对y_{new}做出较为准确的预测。

3.1.2　最小二乘

在线性回归中最常用的是均方误差作为其损失函数,基于最小二乘的方法求解回归问题参数$\boldsymbol{w}^*$和b^*:

$$\frac{1}{N}\sum_{i=1\ldots N}\left(f_{w,b}\left(\boldsymbol{x}_i\right)-y_i\right)^2$$

(3.2)

□□□□,□□□□□□□□□□□□□□□□□□(objective function),□□□□“□□”□□□3.2□,($f_{\mathbf{w},b}(\boldsymbol{x}_i)$-$y_i$)^2□□□□(loss function),□□□□□□□□□□□□□□□□□,□3.2□□□□□□□□□□(squared error loss)□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□[□□□□□(cost function)]□□□□□□□□□□□□□□□□,□□□□□□(empirical risk)□□□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□(quadratic function)□?□□□□□□□y_i□□□$f(\boldsymbol{x}_i)$□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□y_i□$f(\boldsymbol{x}_i)$□□□□□□□□□□□,□□□□□(binary loss)(y_i□$f(\boldsymbol{x}_i)$□□□□□□1,□□□0)□□□□□?

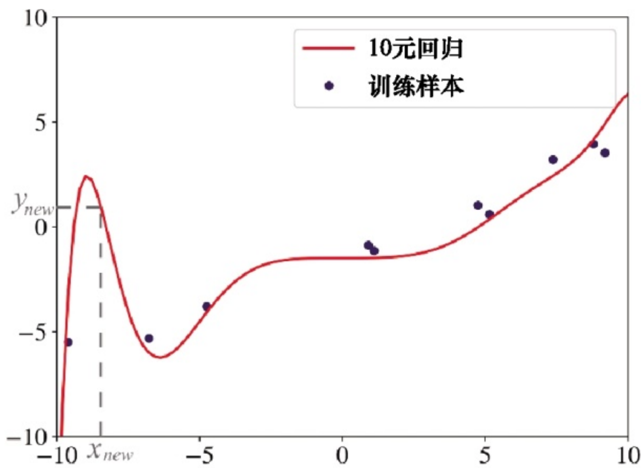
□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□?□□,□□□□□□□□,□□□□□□□□□

□□,□□□□□□□□□□□□□□

- □□□□□□□□□□□□□,□□□□□□□□□
- □□□□□□□□□□,□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□?□□□□□□□,□□□□□□□□□□□□(overfitting)□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

□3.2□□□□□□□□□□□□□□□□□□□□□□□□(□□)□□3.1□□□□□□□□□□□□,□□□□□□□(□□)□□□10□□□□□□(polynomial regression)□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□5□□□□□□□□□□□□□□□□□□□



□3.2 □□□□□

□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□?□□□□□□□□□□□□□□□□□□?1805□,□□□□□□□□□·□□·□□□(Adrien-Marie Legendre)□□□?□□□□□□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□□□□□□□□□□□□□□□□□(closed form solution)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[□□□□□□□□□□□□□□□□□(gradient descent)],□□□□□□,□□□□□□

以此类推,我们可以求出一个高维平面来将数据一分为二。在第3章和4章中会详细说明此原理,这里只简单地提一下。

那么,训练分类器的过程是什么样的呢?这将在第3.2节中,逻辑回归的模型中说明,可以找到一个损失0[2]尽可能小的一组参数,我们把它记为$\boldsymbol{w}^*$和b^*。

3.2 逻辑回归模型

逻辑回归模型(logistic regression)[3]虽然叫做回归模型,但是它是用来做分类的,而且它是一个二分类的模型,原理就是去找一个合适的超平面。

下面从模型的公式和损失函数两方面阐述逻辑回归模型到底是如何进行分类的。

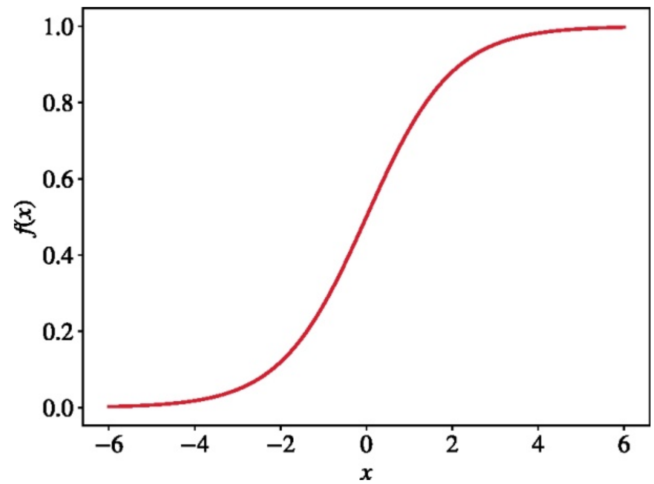
3.2.1 模型公式

在线性分类器里面,对一个样本点$\boldsymbol{x}_i$有一个取值为y_i的标签。而标签y_i只有两种,所以我们要有一个函数能够将线性模型$\boldsymbol{w}\boldsymbol{x}_i+b$的连续输出值转化为离散的值,与$y_i$的取值保持一致。

在做二分类问题时,我们需要的并不是一个输出值,而是一个二分类其中一类的概率值,所以函数的值域不能小于0、而且不能大于1,最好是输出值恰好在区间(0,1)内。基于上面的这种想法,我们希望$\boldsymbol{x}$越大越接近0时,函数随$\boldsymbol{x}$的增大而增;而且,$\boldsymbol{x}$越大越接近,那就是标准逻辑函数(standard logistic function),也称为sigmoid函数,该函数的具体形式可以写成:

$$f(x)=\frac{1}{1+\mathrm{e}^{-x}}$$

□□,e□□□□□□□□,□□□□□(Euler's Number)□□□□□□□exp(x)□□ e^x□sigmoid□□□□□□3.3□□□



□3.3 □□□□□□□□

□□□□□□□□□□:

$$f_{\boldsymbol{w},b}\left(\boldsymbol{x}\right) \stackrel{\text{def}}{=} \frac{1}{1+\mathrm{e}^{-(\boldsymbol{w}\boldsymbol{x}+b)}} \tag{3.3}$$

□□□□□□□□□□□□$\boldsymbol{wx}+b$□

□□□□□□□□□□□□,□□□□□□□□□□□□□□□:□□□□□□□□□$\boldsymbol{w}$□b□,□□□□$f(\boldsymbol{x})$□□□□□y_i□□□□□□□□□,□□□□□□□□□□□0.5,□□□□$\boldsymbol{x}$□□□□;□□□□□□□□□,□□□□□□□□□□□□□□□□□□□5□□□□□□□□□□□□□□

□□□,□□□□□□□□□$\boldsymbol{w}^*$□b^*□?□□□,□□□□□,□□□□□□□□□,□□□□□□(Mean Squared Error,MSE)□

3.2.2 □□□□

□□□□□□□□,□□□□□□□□□□□□□□□□(likelihood)□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□□($\boldsymbol{x}_i$,y_i)□□□,□□□□□□□□□(□□□)□□□□□□□□$\hat{\boldsymbol{w}}$□$\hat{b}$□□□,□□□3.3□□□□□$\boldsymbol{x}_i$□□□$f_{\hat{\boldsymbol{w}},\hat{b}}$□□□,□□□□□□□0<$p$<1□□□□□$y_i$□□□,□□□□□□,$y_i$□□□□□□□□□$p$□□□□,□□$y_i$□□□,□$y_i$□□□□□□□□1-$p$□

□□□□□□□□□□(optimization criterion)□□□□□□(maximum likelihood)□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□:

$$L_{\boldsymbol{w},b} \stackrel{\text{def}}{=} \prod_{i=1\ldots N} f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}\left(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i)\right)^{(1-y_i)} \tag{3.4}$$

□□□$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i))^{(1-y_i)}$□□□□,□□□□□□□□□□:□$y_1$=1□□$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)$□□,□□□(1-$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)$)□□□□□□□□,□□$y_1$=1,□(1-$y_1$)=0□□□□□□□□□□0□□□□1,□□(1-$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)$)$^{(1-y_i)}$□□1□□□,□□$y_1$=0,$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}$□□1□

□□□□□□□□,□□□□□□□□□□□□□□□Π□□□□□□□□□□□□□□Σ□□□□□□,□□□N□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□□□,□□□□□□□)□□□□□□□□□□□(log-likelihood)□:

$$\log L_{\boldsymbol{w},b} \stackrel{\text{def}}{=} \ln\left(L_{\boldsymbol{w},b}(\boldsymbol{x})\right) = \sum_{i=1}^{N} y_i \ln f_{\boldsymbol{w},b}(\boldsymbol{x}) + (1-y_i)\ln\left(1-f_{\boldsymbol{w},b}(\boldsymbol{x})\right)$$

□□ln□□□□□□□□□(strictly increasing function),□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□(gradient descent)□□□□□□,□□□□□□□□□

3.3 □□□□□

□□□(decision tree)□□□□□□□□□□□□□(graph)□□□□□□□□□,□□□□*i*□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□□□□□□□(leaf node)□,□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□,□□□□□□□□□□□□□□□□

3.3.1 □□□□

□□□□□□□□:□□□□□{0,1}□□□□□□□□□□□□,□□□□□□□□□□□□□□□

3.3.2 □□□□

□□□□□□□□□□□□□□□□,□□□□□□□□□□ID3□□□□

□□□□□□□□□□□□□□□:

$$\frac{1}{N}\sum_{i=1}^{N} y_i \ln f_{\mathrm{ID3}}\left(\boldsymbol{x}_i\right)+\left(1-y_i\right)\ln\left(1-f_{\mathrm{ID3}}\left(\boldsymbol{x}_i\right)\right)$$

(3.5)

□□,f_{ID3}□□□□□□□

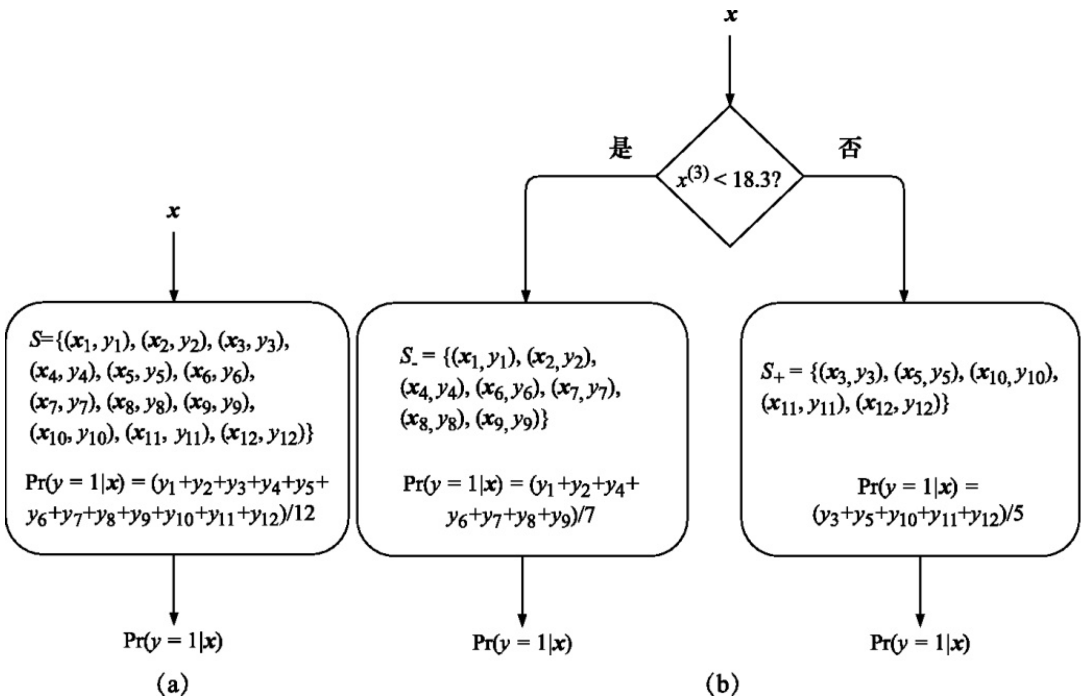
□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□(parametric model),□f_{w^*,b^*};□ID3□□□□□□□□□□□□□□□□□□□(nonparametric model),□$f_{\mathrm{ID3}}(\boldsymbol{x}) \stackrel{\mathrm{def}}{=} \Pr(y=1|\boldsymbol{x})$□

ID3□□□□□□□□□□□S□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□,$S \stackrel{\mathrm{def}}{=} \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$□□□□□□□□□□□:

$$f_{\mathrm{ID3}}^{S} \stackrel{\mathrm{def}}{=} \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} y$$

(3.6)

□□□□□□□□□***x***□□□,$f_{\mathrm{ID3}}^{S}(\boldsymbol{x})$,□□□□□□□□3.4(a)□,□□□□□□□□□12□□□□□□□□



□3.4 □□□□□□□□□□□□□□

□:□□□□S□□12□□□□□□(a)□□,□□□□□□□□□□□□,□□□□□□□□□□□□□(b)□□□□□□□□□□□□□□□□□□□□□□□,□□□□3□□□□18.3,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□,□□□□□□□□□□j=1,…,D□□□□□t,□□S□□□□□□□□:
$S_- \stackrel{\text{def}}{=} \{(\boldsymbol{x}, y) | (\boldsymbol{x}, y) \in S, x^{(j)} < t\}$□$S_+ \stackrel{\text{def}}{=} \{(\boldsymbol{x}, y) | (\boldsymbol{x}, y) \in S, x^{(j)} \geqslant t\}$□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□(j,t)□S_+□S_-□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□S_+□S_-(□□□□□□□□□□□□□□□□□□□□□□□□□□□)□□3.4(b)□□□□□□□□3.4(a)□□□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□?□ID3□,□□□□(entropy)□□,□□□□□□□□□□□□□□S□□□□□□□:

$$H(S) \stackrel{\text{def}}{=} -f_{\text{ID3}}^{S} \ln f_{\text{ID3}}^{S} - \left(1 - f_{\text{ID3}}^{S}\right) \ln \left(1 - f_{\text{ID3}}^{S}\right)$$

□□□□□□□□□j□□□□□t□□□□□□□□□,□□□□□$H(S_-,S_+)$□□□□□□□□□□□

□□

$$H\left(S_{-}, S_{+}\right) \stackrel{\text{def}}{=} \frac{\left|S_{-}\right|}{|S|} H\left(S_{-}\right)+\frac{\left|S_{+}\right|}{|S|} H\left(S_{+}\right)$$

(3.7)

□ID3□□□□□,□□□□□□,□□□□3.7□□□,□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□,□□□□□□□□□:

- □□□□□□□□□□□□□□(□3.6)□□□□□
- □□□□□□□□□□□□□
- □□□□□□□□□□□ε□(□□□□□□□□[4])□
- □□□□□□□□□□□(□□□□□□□)□

□□□□□□□,□□□□□□□□□□□□□□(□□□□□□□□□),ID3□□□□□□□□□□□□

□□□□□□□□□□(backtracking)□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□C4.5□□□ID3,□□□□□□:

- □□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□(bottom-up)□“□□”(pruning)□□□□□□□□□□

□□□□□□□□:□□□□□□□□,□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□

□□

□□□□□□□□□□□□□:□*S*□□□□□□□□□□□□□□,□□□□□□0;□□□□□□□□□□□□□,□□□□□1,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

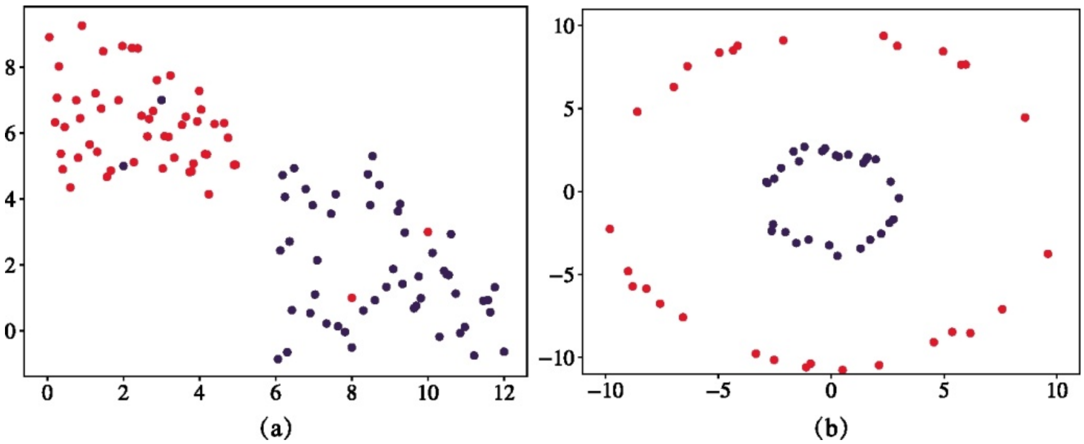
3.4 □□□□□

□□1□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□:

1□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□?

2□□□□□□□□□□□□□□,□□□□□□□□□□□□□(higher-order polynomial)□□,□□□?

□□□□□□□□3.5□□□□□□□,□□□□□□□(□□□□□□□□□□□),□□□□□□□□□□□□□□3.5(b)□,□□□□□□□□,□□□□□□□□



□3.5　□□□□□□□□□

□□□□SVM,□□□□□□□□□□□□:

$$
\begin{aligned}
\boldsymbol{w}\boldsymbol{x}_i - b \geqslant +1, &\quad 当\ y_i = +1\ 时 \\
\boldsymbol{w}\boldsymbol{x}_i - b \leqslant -1, &\quad 当\ y_i = -1\ 时
\end{aligned}
$$

(3.8)

□□,□□□□□$\|\boldsymbol{w}\|$□□□,□□□□□□□□□□□□□□□□□□□□□□□$\|\boldsymbol{w}\|$□□□□□□ $\frac{1}{2}\|\boldsymbol{w}\|^2$□□□□□□□□□,□□□□□□□□□□□□□□□(quadratic programming)□□□□□□□,SVM□□□□□□□□:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2, \quad □□, y_i(\boldsymbol{x}_i\boldsymbol{w} - b) - 1 \geqslant 0, i = 1, \ldots, N$$

(3.9)

3.4.1　□□□□

□□□□SVM□□□□□□□□□□□,□□□□□□□□□□(hinge loss)□□:max(0,1-y_i($\boldsymbol{wx}_i$-b))□

□□3.8□□□□□□□□,□□□□□0□□□□□,□□$\boldsymbol{wx}_i$□□□□□□□□□□□□□,□□□□□0□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□,□□□□□□□□□□□□□:

$$C\|\boldsymbol{w}\|^2+\frac{1}{N}\sum_{i=1}^{N}\max\left(0,1-y_i\left(\boldsymbol{w}\boldsymbol{x}_i-b\right)\right)$$

□□□□□□□□□□□□,□□□□□□□□$\boldsymbol{x}_i$□□□□□□□□□□□□C□□□□□□□□□□□□ID3□□□□□ϵ□d,C□□□□□□□□□□□□□□□□□SVM□□□□□(soft-margin)SVM□

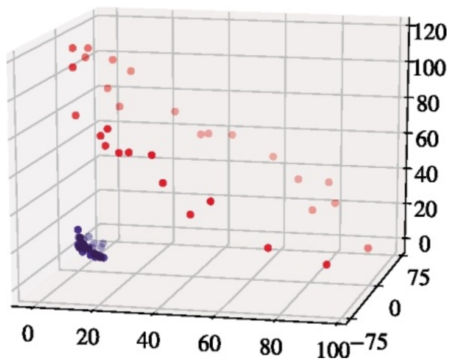
□□□□□□,□C□□□□□,□□□□□□□□□□□□□□□□□SVM□□□□□□□□□□,□□□□□□□□□□□□C□□□□□□□□□□□□□□□□□□□□□□,SVM□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,C□□□□□□□□□□□□(□□□□□□□)□□□□□(□□□)□□□□□□□

3.4.2 □□□□□□□

SVM□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□SVM□□□□□□,□□□□□□□□□(implicitly)□□□□□□□□□□□□□□□□□□□□□(kernel trick)□

□□□□□□□□□3.6□□,□□□□□□□□□(mapping)ϕ:***x****a*ϕ(***x***)□□□□□

□□□□□□□□□□□□□□□□□□□□,ϕ(***x***)□□□□***x***□□□□□□□□3.5(□)□□□□□□□□,□□□□□□***x***=[*q*,*p*]□□□□□□□□(□□3.6)□,$\phi([q,p]) \stackrel{\text{def}}{=} (q^2, \sqrt{2}qp, p^2)$□□□,$\cdot^2$□□□□□□□□,□□□□□□□□□□□□□□□□□



□3.6　□□□□□□□□□□□,□3.5(b)□□□□□□□□□□□

□□,□□□□□□□□□□□□ϕ□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□SVM□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[kernel function,□□□□(kernel)]□□□□□□□□□,□□□□□□□SVM□□□□□□□□□***w***□*b*□□□□□

□□□□□3.9□□□□□□□□□□□□(method of Lagrange multiplier)□□□□□□3.9□□,□□□□□□□□□□□□□:

$$\max_{\alpha_1 \ldots \alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} y_i \alpha_i (\boldsymbol{x}_i \boldsymbol{x}_k) y_k \alpha_k$$

□□□ $\sum_{i=1}^{N} \alpha_i y_i = 0$ □□ $\alpha_i \geq 0, i=1,\ldots,N$

□□,α□□□□□□□□□,□□□□□□□□□□□□□□(convex quadratic optimization problem),□□□□□□□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□,□$\boldsymbol{x}_i\boldsymbol{x}_k$□□□□□□□□□□□□□□□□□□□,□□□$\boldsymbol{x}_i$□$\boldsymbol{x}_k$□□□□$\phi(\boldsymbol{x}_i)$□$\phi(\boldsymbol{x}_k)$□□□□□□□□□□□□□□

□□,□□□□$\boldsymbol{x}_i\boldsymbol{x}_k$□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□(q_1,p_1)□(q_2,p_2)□□□□($q_1^2,\sqrt{2}q_1p_1,p_1^2$)□($q_2^2,\sqrt{2}q_2p_2,p_2^2$)□□□□□□($q_1^2q_2^2+2q_1q_2p_1p_2+p_1^2p_2^2$),□□□□□$(q_1,p_1)$□$(q_2,p_2)$□□□$(q_1q_2,p_1p_2)$,□□□□□□□□□□□□($q_1^2q_2^2+2q_1q_2p_1p_2+p_1^2p_2^2$)□

□□□□□□□□□□□□,□□□□$k(\boldsymbol{x}_i,\boldsymbol{x}_k)\stackrel{\text{def}}{=}(\boldsymbol{x}_i\boldsymbol{x}_k)^2$□□□□□□,□□□□□□□□□(Radial Basis Function,RBF)□:

$$k\left(\boldsymbol{x},\boldsymbol{x}'\right)=\exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2}{2\sigma^2}\right)$$

□□,$\left\|\boldsymbol{x}-\boldsymbol{x}'\right\|^2$□□□□□(Euclideandistance)□□□□□□□□□□□□□□:

$$\begin{aligned} d\left(\boldsymbol{x}_i,\boldsymbol{x}_k\right) &\stackrel{\text{def}}{=} \sqrt{\left(x_i^{(1)}-x_k^{(1)}\right)^2+\left(x_i^{(2)}-x_k^{(2)}\right)^2+\ldots+\left(x_i^{(D)}-x_k^{(D)}\right)^2} \\ &= \sqrt{\sum_{j=1}^{D}\left(x_i^{(j)}-x_k^{(j)}\right)^2} \end{aligned}$$

RBF[illegible]σ,[illegible]

3.5 k[illegible]

k[illegible](k-Nearest Neighbor,kNN)[illegible],[illegible]kNN[illegible]***x***[illegible],kNN[illegible]***x***[illegible]k[illegible],[illegible],[illegible]

[illegible],[illegible](consine similarity)[illegible]:

$$s\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right) \stackrel{\text{def}}{=} \cos\left(\angle\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right)\right) = \frac{\sum_{j=1}^{D} x_i^{(j)} x_k^{(j)}}{\sqrt{\sum_{j=1}^{D}\left(x_i^{(j)}\right)^2}\sqrt{\sum_{j=1}^{D}\left(x_k^{(j)}\right)^2}}$$

[illegible]0[illegible],[illegible],[illegible]1[illegible],[illegible]0[illegible],[illegible]-1[illegible],[illegible]-1[illegible](Chebychev distance)[illegible](Mahalanobis distance)[illegible](Hamming distance)[illegible]k[illegible],[illegible]([illegible])[illegible]10[illegible]

[1] □□□$y_i \in \mathbb{R}$□□□y_i□□□□□□□□□□,$\mathbb{R}$□□□□□□□□□,□□□□□□□□□□□□□□
□□

[2] □□□□□□□□□□□□,□□□□□□□0□□□□,□□□□□□□□□□0□□□□□□□,□□□
□□□□□□□□□□□

[3] □□□:□□□□□□,□□□□□□□□□□□□□□□□□□□□“□□□□”□“□□□□□□”□

[4] □□5□,□□□□□□□□□□□

[illegible]4[illegible] [illegible]

4.1 [illegible]

[illegible],[illegible]3[illegible]:

- [illegible]
- [illegible]([illegible])[illegible]
- [illegible]

[illegible],[illegible]([illegible],[illegible],[illegible]),[illegible]([illegible]k[illegible])[illegible]k[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible]

[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible]SVM[illegible],[illegible],[illegible]SVM,[illegible],[illegible],[illegible],[illegible],[illegible]

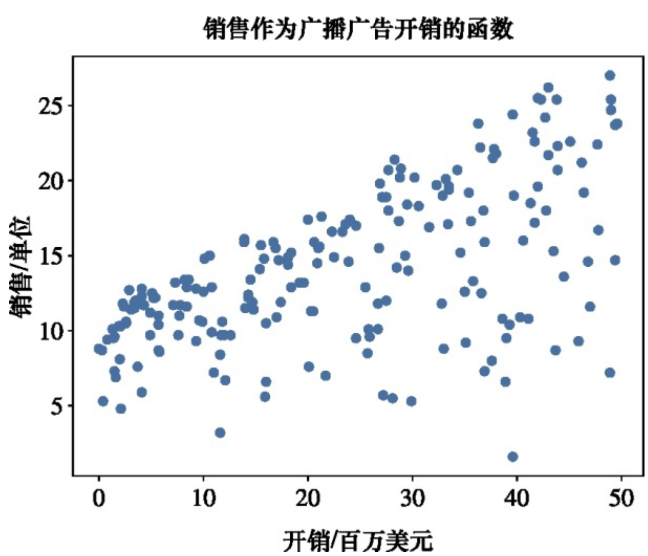
□□□□□□□□□□□□□□□□□□□□

4.2 □□□□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□[1]□□□□Python□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□:w□b□□□□□□□□□□□□□□□:□□□□□□□□□□$w^{(1)}$,$w^{(2)}$□b,□□□□□□$w^{(1)}$,$w^{(2)}$,$w^{(3)}$□b,□□□□□

□□□□□□□□□,□□□□□□□□□□□□(□□□□□□□□□□□)□□□□□□□□□:□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□

□□□200□□□□□,□□□200□□□□,□□□□□(x_i,y_i)=(□□,□□)□□□□□□□□□□□4.1□□□□□



□4.1 □□□

□:y□□□□□□□(□□□□□□□□□),x□□□□□□□□,□□□□□□□□□□(□□□□)□

□□	□□/□□□□	□□/□□
1	37.8	22.1
2	39.3	10.4
3	45.9	9.3
4	41.3	18.5
...	...	...

□□□□□□,□□□□□□□□□□□:$f(x)=\boldsymbol{wx}+b$□□□□□□□w□b□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□w□b□:

$$l \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N}\left(y_i-(wx_i+b)\right)^2$$

□□□□□□□□□□□□□□□□□:

$$\frac{\partial l}{\partial w}=\frac{1}{N}\sum_{i=1}^{N}-2x_i\left(y_i-(wx_i+b)\right);$$
$$\frac{\partial l}{\partial b}=\frac{1}{N}\sum_{i=1}^{N}-2\left(y_i-(wx_i+b)\right)$$

(4.1)

□□$(y_i-(wx+b))^2$□w□□□□,□□□□□□□□□□□□□□,$f=f_2(f_1)$□□□□□□□,$f_1=y_i-(wx+b)$,□$f_2 = f_1^2$□□□f□w□□□□,□□□f□f_2□□□□,□□$2(y_i-(wx+b))$(□□□□□,□□□□□□$\frac{\partial f}{\partial x}x^2 = 2x$)□□□,□□$y_i-(wx+b)$□$w$□□□□$-x$□□□□□□,□$\frac{\partial l}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i-(wx_i+b))$□□□□□,□$l$□$b$□□□□$\frac{\partial l}{\partial b}$□

□□□□□□□(epoch)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□[2]□□$w \leftarrow 0$□$b \leftarrow 0$□□□□□$\frac{\partial l}{\partial w}$□$\frac{\partial l}{\partial b}$□□□□$\frac{-2}{N}\sum_{i=1}^{N} x_i y_i$□$\frac{-2}{N}\sum_{i=1}^{N} y_i$□□□□□□,□□□□□□□$w$□$b$□□□□□□□□□□□$\alpha$□□□

$$w \leftarrow w - \alpha\frac{\partial l}{\partial w};$$
$$b \leftarrow b - \alpha\frac{\partial l}{\partial b}$$

(4.2)

□□□□□□□□□(□□□□)□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□[3]□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□□□□□□□(□□□□□□)□□□□□□□□(□□□□□),□□□□□□□□,□□□□□□□□□□□

□□□□□□,□□□□4.1□□□w□b□□□□□□□□;□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□w□b□□□□□□□□□□□□□□□,□□□□□□□□□

□□□□□□□□□□□□Python□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□Python□□□□□□□□□□□□□

□□□□□□,□□□□w□b□□□□□□□□

```
def update_w_and_b(spendings, sales, w, b, alpha):
    dl_dw = 0.0
    dl_db = 0.0
    N = len(spendings)

    for i in range(N):
        dl_dw += -2*spendings[i]*(sales[i] - (w*spendings[i] + b))
        dl_db += -2*(sales[i] - (w*spendings[i] + b))

    # update w and b

    w = w - (1/float(N))*dl_dw*alpha
    b = b - (1/float(N))*dl_db*alpha

    return w, b
```

□□□for□□□□□□□□□□□□□□□□

```
def train(spendings, sales, w, b, alpha, epochs):
    for e in range(epochs):
        w, b = update_w_and_b(spendings, sales, w, b, alpha)

        # log the progress
        if e % 400 == 0:
            print("epoch:", e, "loss: ", avg_loss(spendings,
                sales, w, b))
    return w, b
```

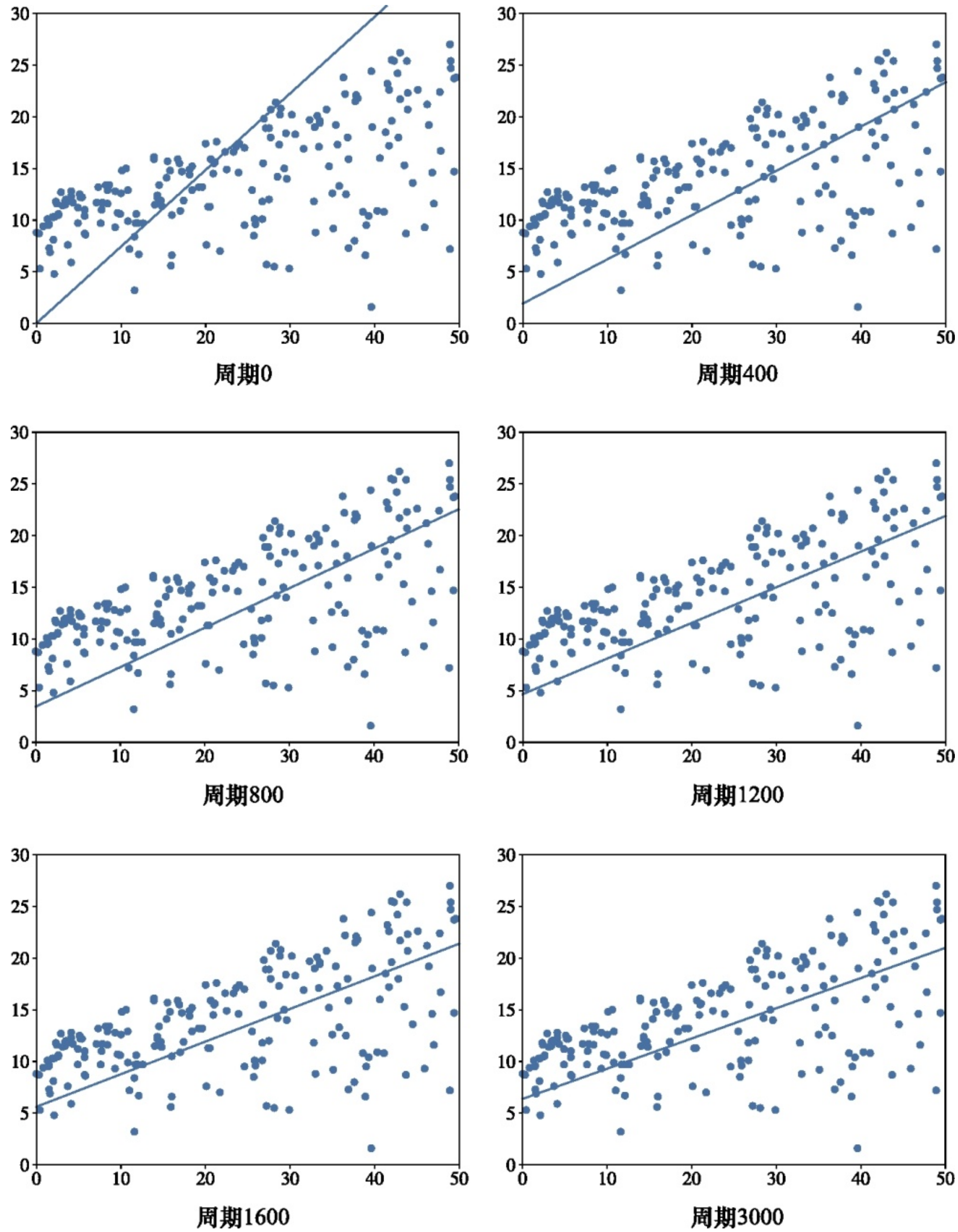
□□train□□□□avg_loss□□□□□□□□□□□□,□□□□□□:

```
def avg_loss(spendings, sales, w, b):
    N = len(spendings)
    total_error = 0.0
    for i in range(N):
        total_error += (sales[i] - (w*spendings[i] + b))**2
    return total_error / float(N)
```

□□□□□□□□α=0.001□w=0.0□b=0.0,□□□□□1 500□□train□□,□□□□□□□□□(□□□□□□□)□

```
epoch:  0 loss: 92.32078294903626
epoch:  400 loss: 33.79131790081576
epoch:  800 loss: 27.9918542960729
epoch:  1200 loss: 24.33481690722147
epoch:  1600 loss: 22.028754937538633
...
epoch:  2800 loss: 19.07940244306619
```

□□□□□□,□□train□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□
4.2□□□



[illegible]4.2 [illegible]

[illegible],[illegible] w [illegible] b [illegible],[illegible]:

```
31  def predict(x, w, b):
32      return w*x + b
```

□□□□□□□□□

```
33  w, b = train(x, y, 0.0, 0.0, 0.001, 15000)
34  x_new = 23.0
35  y_new = predict(x_new, w, b)
36  print(y_new)
```

□□□□□□□13.97□

□□□□□□□□□α□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□
□□□□□□□□□□□□□□□□□□

□□□□□□□□□(minibatch Stochastic Gradient Descent,minibatch SGD)□□□□□□□□□□□,□□□□□□□□□□□(□□)□□□□,□□□□□□□□SGD□□□□□□“□□”□□□,Adagrad□□□□□□□□□,□□□□□□□□□□□□□□□□□□α:□□□□□□□α□□,□□□□□□□(Momentum)□□□□□SGD□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□SGD□□□,□RMSprop□Adam□

□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□(□□□□□□□□□□)□

4.3 □□□□□□□□□□□

□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(library),□□□□□□□□□□(open sourced)□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□scikit-learn,□Python□C□□□□□□□□□scikit-learn□□□□□□:

```python
def train(x, y):
    from sklearn.linear_model import LinearRegression
    model = LinearRegression().fit(x,y)
    return model

model = train(x,y)

x_new = 23.0
y_new = model.predict(x_new)
print(y_new)
```

□□□□□□□□13.97□□□□□□□?□□□□□LinearRegression(□□□□)□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□LogisticRegression(□□□□)□□SVC(□□□□□□scikit-learn□□□□),DecisionTreeClassifier(□□□)□NearestNeighbor(k□□)□□□□□scikit-learn□□□□□□□□□

4.4 □□□□□□□

□□□,□□(□□,SVM□□C,ID3□□ϵ□d);□□□□□(□□□□□)□□□□□,□□α□

□□□□,□□□□□,□□□□□□□□□□□□□,□□□□□□□“□□”□□,□□□□“□”“□”“□”,□□□□□□□□□□SVM□□□□□□□□□k□□(□□□□□□□□□□□□□□)□□□□□□□□□,scikit-learn□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□,□□SVM,□□□,□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□)□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□

[illegible],[illegible]SVM[illegible]k[illegible],[illegible],[illegible],[illegible]0[illegible]1[illegible],[illegible]

[illegible]([illegible]SVM)[illegible],[illegible]([illegible]SGD[illegible]/SGD[illegible]scikit-learn[illegible]PassiveAggressiveClassifier/PassiveAggressiveRegressor)[illegible],[illegible]

[illegible],[illegible]([illegible]SVM[illegible]k[illegible])[illegible];[illegible],[illegible]

[illegible],[illegible]

---

[1] [illegible],[illegible],[illegible],[illegible]

[2] [illegible]([illegible])[illegible],[illegible]([illegible]0[illegible]0[illegible]),[illegible]

[3] [illegible]

[illegible]5[illegible] [illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]scikit-learn[illegible]model=LogisticRegression().fit(x,y)[illegible]

5.1 [illegible]

[illegible],[illegible]:“[illegible]5[illegible]”[illegible],[illegible],[illegible](dataset)[illegible]

[illegible],[illegible]1[illegible],[illegible](labeled example)[illegible],[illegible] $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ [illegible] N [illegible] $\boldsymbol{x}_i$ [illegible](feature vector)[illegible] j=1,…,D [illegible](feature),[illegible] $x^{(j)}$ [illegible]

[illegible](feature engineering)[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible](session)[illegible]([illegible]),[illegible]

(request)□□□□□□□(response time)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(informative)□□□:□□□(predictive power)□□□□□□,□□□□(session)□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

5.1.1 □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□“□□”□“□□□□□□”,□□□□□□□□□□□□□□□□□□□□□

□□□□□□□“□□”□□,□□3□□□□□“□”“□”“□”,□□□□□□□□□□□□□□□3□□□□□□□□:

$$
\begin{aligned}
\text{红} &= [1,0,0] \\
\text{黄} &= [0,1,0] \\
\text{绿} &= [0,0,1]
\end{aligned}
\tag{5.1}
$$

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1□□□□□2□□□□□3□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□[1]□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□

5.1.2 □□

[illegible](binning)[illegible](bucketing)[illegible],[illegible],[illegible],[illegible]:[illegible]0~5[illegible],6~10[illegible],11~15[illegible],[illegible]

[illegible],[illegible]j=4[illegible],[illegible]3[illegible],“[illegible]1”“[illegible]2”“[illegible]3”,[illegible]j=123[illegible]j=124[illegible]j=125[illegible],[illegible]x[illegible]$x_i^{(4)} = 7$,[illegible]$x_i^{(124)} = 1$;[illegible]$x_i^{(4)} = 13$,[illegible]$x_i^{(125)} = 1$,[illegible]

[illegible],[illegible]“[illegible]”,[illegible],[illegible]

5.1.3 [illegible]

[illegible](normalization)[illegible],[illegible][-1,+1][illegible][0,1][illegible]

[illegible],[illegible]350[illegible]1450,[illegible]350,[illegible]110,[illegible][0,1][illegible]

[illegible]:

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}}$$

[illegible],$\min^{(j)}$[illegible]$\max^{(j)}$[illegible]j[illegible]

为什么会出现这种情况?原因是,特征值的量级不同会导致模型参数的量级不同,进而影响到梯度下降的收敛速度。如图4所示,在特征值量级不同的情况下,梯度下降会偏离最优路线。图中的参数为$w^{(1)}$和$w^{(2)}$等,在这里为简化起见$w^{(1)}$和$w^{(2)}$分别对应于特征$x^{(1)}$取值范围[0,1 000],而$x^{(2)}$的取值范围[0,0.000 1],可以看出梯度下降会偏离最优路线。

因此,在进行梯度下降算法之前要先对数据进行处理,这也被称作特征缩放,就是让各个特征具有相同的(或相近的)量级。

5.1.4 标准化

标准化(standardization)又称**z**值归一化(z-score normalization),它可以将数据转换成均值μ=0和σ=1的标准正态分布的数据。其中,μ是均值(对特征中的所有样本求平均值),σ是它们的标准差。

计算公式的标准分数(或z分数)计算公式如下:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}$$

标准化的公式很简单,但是做这样的变换为什么能提高梯度下降算法的收敛速度呢?原因是对数据进行标准化处理之后,数据的均值变成零,标准差为一,所有特征具有相同的量级,这样梯度下降的时候收敛就会更快一些。

在需要使用标准化的算法里面,通常包括以下几种:

- 线性回归,标准化可以提升梯度下降算法的收敛速度。
- 基于距离计算的算法需要对各个特征(即样本的维度)做标准化,如聚类算法等。
- 对于使用正则化惩罚项的模型中的参数(即权重),它们也需要进行标准化;否则在正则化惩罚参数的时候,量级大的特征“惩罚”得更少而不公平。

- [illegible],[illegible]

[illegible],[illegible]

5.1.5 [illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]

[illegible]:

- [illegible]([illegible],[illegible])[illegible]
- [illegible]([illegible])[illegible]
- [illegible]

5.1.6 [illegible]

[illegible],[illegible]:

$$\hat{x}^{(j)} \leftarrow \frac{1}{M}\sum_{i=1}^{N} x_i^{(j)}$$

[illegible]$M<N$,M[illegible]j[illegible],[illegible]j[illegible]

[illegible],[illegible][0,1],[illegible]

[illegible]2[illegible]-1[illegible],[illegible],[illegible],[illegible]

将所有连续型特征的取值缩放到一个区间,比如可以选择缩放到区间[-1,1],或者缩放到以数据均值为0、方差为单位方差的正态分布上,以此类推。

采用数据补全技术时,可以对缺失的数据特征创建一个回归模型,则对每一条数据样本[$x_i^{(1)}$,$x_i^{(2)}$,...,$x_i^{(j-1)}$,$x_i^{(j+1)}$,...,$x_i^{(D)}$]构建一个特征$\hat{\boldsymbol{x}}_i$,令$\hat{y}_i$←$x^{(j)}$的取值,j为具有缺失值的特征,可以使用该回归模型将$\hat{\boldsymbol{x}}_i$作为$\hat{y}_i$的预测值构建样本,再用该模型进行预测以补全特征j的值,即可得到样本集($\hat{\boldsymbol{x}}$,$\hat{y}$)。

最后,如果数据集很大,并且是一个高维度的数据集,可以通过增加数据维度的方式,为每个特征增加一个二元指示特征。假设每个样本的特征维度为D的数据集中,特征j=12的取值缺失。为每一个样本x中,增加一个新的特征j=D+1。如果特征12的值存在,则该特征值为1;否则设置为0。对于缺失值,可以将该缺失的特征取值替换为0或任意一个数值。

预测阶段时,如果样本数据不完整,应该使用与训练阶段相同的数据补全技术来填补缺失的特征。

选择哪种数据补全技术时,需要尝试多种技术,并使用每种技术构建多个模型后进行对比,从中选择对数据集效果最好的技术。

5.2 学习算法选择

选择机器学习的算法可能是一项艰巨的任务。如果有足够的时间,则可以尝试所有学习算法。然而,通常来说,解决问题的时间是有限的。在开始研究问题之前,可以考虑一下以下几个问题,然后根据这些问题的答案,缩小可选算法的范围,并在数据上进行尝试。

• 可解释性

对于待选择的模型,该模型必须要对非技术人员具有可解释性吗?许多精确度高的学习算法都是所谓的“黑盒”模型。这类模型往往在预测时错误很少,但可能很难理解模型为什么会做出这种预测,更难向他人解释。比如神经网络

□□□□□□

□□,□k□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

- □□□vs.□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□?□□□□,□□□□□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□(incremental learning algorithm),□□□□□□□□□□□□□□□□□

- □□□□□□□

□□□□□□□□□□□□□□?□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□(□□□□□□□□□□);□□□□□□□□□□□,□SVM□

- □□□□vs.□□□□

□□□□□□□□□□□□□□□□□□□□□□,□□□□□□?□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

- □□□□□□

□□□□□□□□□□□□?□□□□□□□□□□□□□□□?□□□,□□□□□□SVM□□□□□□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□□□□□□□6□□□7□□□□□□□□□□□□□□

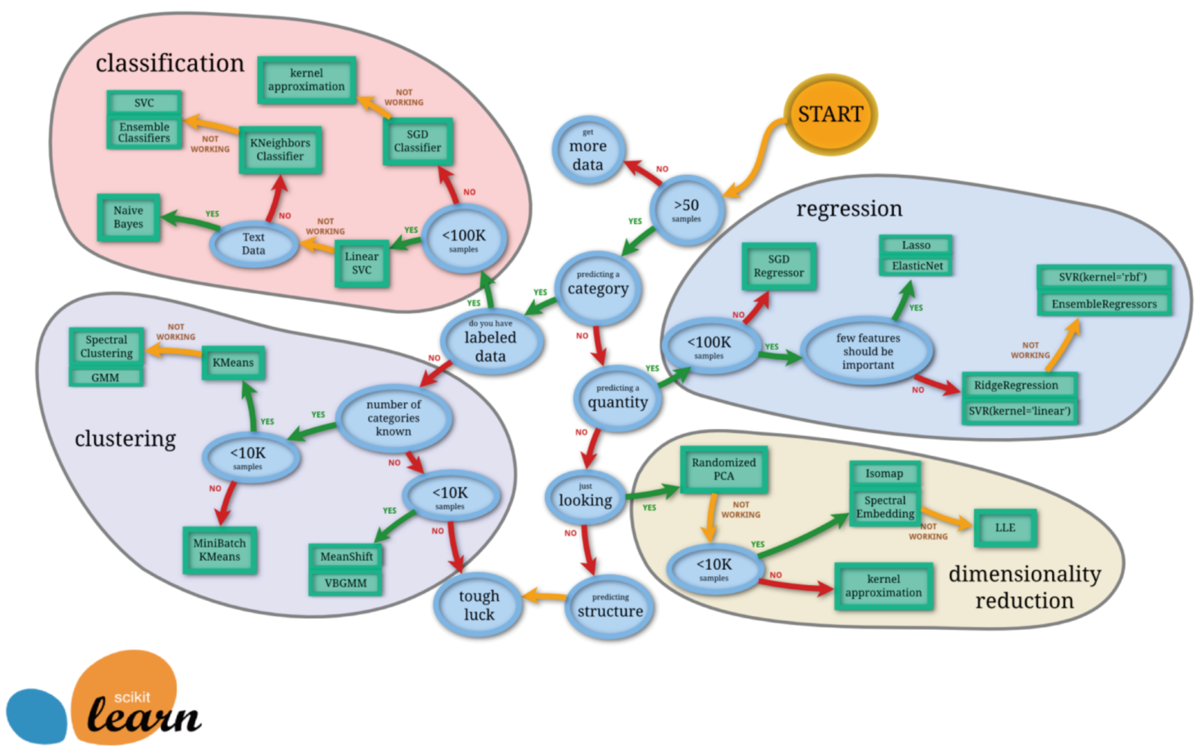
- □□□□

[illegible]?[illegible],[illegible],[illegible]([illegible])[illegible](CPU)[illegible]

- [illegible]

[illegible]?[illegible]?[illegible]SVM[illegible]([illegible])[illegible],[illegible],[illegible]k[illegible],[illegible][2][illegible]

[illegible],[illegible],[illegible]scikit-learn[illegible],[illegible],[illegible]5.1[illegible]



[illegible]5.1 scikit-learn[illegible]

5.3 3□□□□

□□□□,□□□□□“□□□”□“□□□”□□□□□□,□□□□□,□□□□□□□□□□□□□□3□□□□□□:

- □□□□
- □□□□
- □□□□

□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□3□□□:□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□(holdout set)□

3□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□70%□□□□□15%□□□□□□□15%□□□□□□□,□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□95%□□□□□□□□□,□□□□□□□□2.5%□

□□□□□□□,□□□□□□□□□□3□□□,□□□□□□□□□□□?□□□□□:□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□“□□”□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

5.4 □□□□□□□

□□□□□□□,□□□□□□□(bias)□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□(high bias),□□□
□□□□□(underfit)□□
□□□,□□□□□□□:

- □□□□□□□□□□□□□□□(□□,□□□□□□□□□□□□□□)□
- □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□,□□□□□□□□□
□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□
□□□□□,□3□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

□□□(overfitting)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,
□□□□□□□□□□□□□□□□□□□□□□□□□□□□3□□□□□□□□□□□□□□□□□□□□□□,□□□
□□□□:

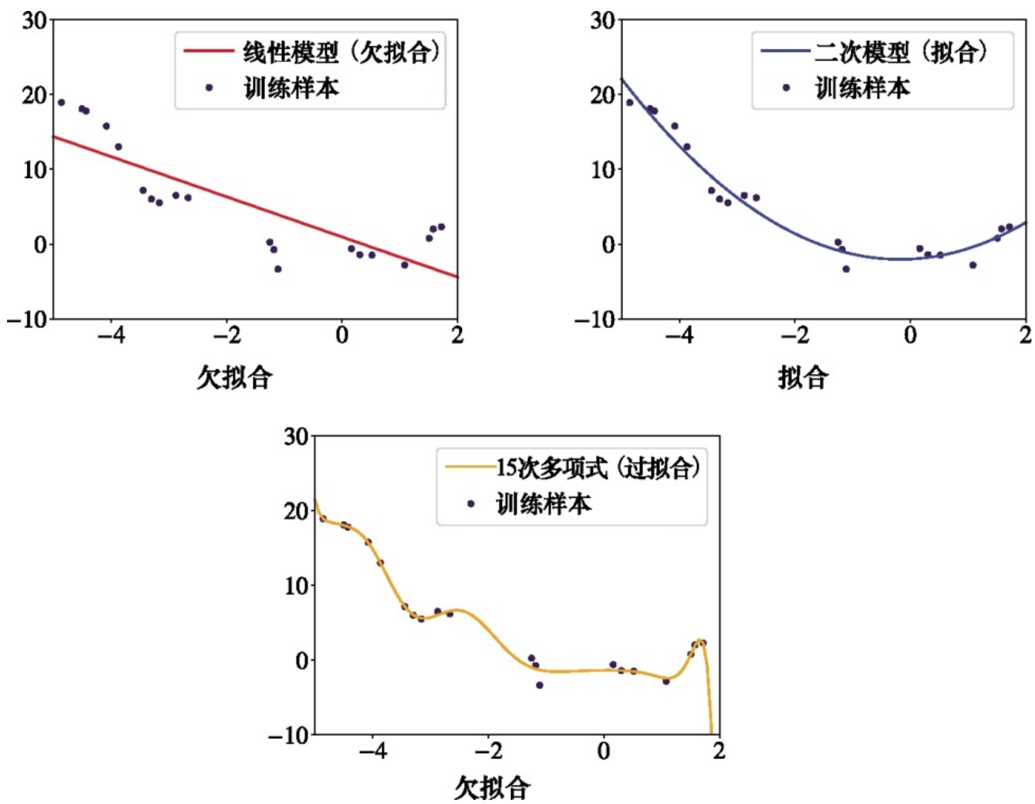
- □□□□□□□□□□□□(□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□)□
- □□□□□,□□□□□□□□

□□□□□□,□□□□□□□□□□□□□□□□□□:□□□(high variance)□□□□□□□
□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□
□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□
□□□□□□□□□□

[illegible],[illegible],[illegible][illegible],[illegible],[illegible],[illegible],[illegible]w[illegible]$w^{(j)}$[illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible]

[illegible]5.2[illegible]



[illegible]5.2 [illegible]([illegible])[illegible]([illegible])[illegible](15[illegible])

[illegible]:

(1)□□□□□□□□□(□□□□□□□□□□□□,□□□□□□□□□□SVM□□RBF□,□□
□□□□□□/□□□□□□□□□□)□

(2)□□□□□□□□□□□(□□,□□□□□□□9□□□□□□□□□□□□□)□

(3)□□□□□□,□□□□□□□□

(4)□□□□□□□□

□□□(regularization)□□□□□□□□□□□□

5.5 □□□

□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□
□,□□□□□□□□□□□□□□□□□□□□□-□□□□(bias-variance tradeoff)□

L1□L2□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

□□□□□□□□□□□□□:

$$\min_{\boldsymbol{w},b} \frac{1}{N} \sum_{i=1}^{N} \left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right) - y_i\right)^2 \tag{5.2}$$

□□L1□□□□□□□□□:

$$\min_{\boldsymbol{w},b}\left[C\left|\boldsymbol{w}\right|+\frac{1}{N}\sum_{i=1}^{N}\left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right)-y_i\right)^2\right]$$

(5.3)

□□, $\left|\boldsymbol{w}\right| \stackrel{\text{def}}{=} \sum_{j=1}^{D}\left|w^{(j)}\right|$,*C*□□□□□□□□□□□□□□□*C*□□0,□□□□□□□□□□□□□□□□□,□□*C*□□□□,□□□□□□□□□□□□□□□$w^{(j)}$□□□□□0□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□*C*□□,□□□□□□□□□,□□□□□□□□□□□□□□6□□,□□□□□□□□

□□L2□□□□□□□:

$$\min_{\boldsymbol{w},b}\left[C\|\boldsymbol{w}\|^2+\frac{1}{N}\sum_{i=1}^{N}\left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right)-y_i\right)^2\right],$$

□ $\|\boldsymbol{w}\|^2 \stackrel{\text{def}}{=} \sum_{j=1}^{D}(w^{(j)})^2$ □

(5.4)

□□□□□,L1□□□□□□□□□(sparse model)□□□□*C*□□□□□,□□□□□□□(□□□□□□□,□□$w^{(j)}$)□□□0□□□,L1□□,□□□□□□□□□□□□□□,□□L2□□□□□□□□□,L2□□□□□□,□□□□□□□□□□□□

□□□□□□□□□(elastic net regularization)□□□□L1□L2□□,L1□L2□□□□□□□□□□□□□□□L2□□□□□(ridge

regularization),[illegible]L1[illegible](lasso)[illegible]

[illegible],L1[illegible]L2[illegible],[illegible]

[illegible]

[illegible],[illegible](dropout)[illegible](batch normalization)[illegible],[illegible]:[illegible](data augmentation)[illegible](early stopping)[illegible]8[illegible]

5.6 [illegible]

[illegible],[illegible]?[illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible](mean model)[illegible],[illegible]

[illegible],[illegible][3](MSE)[illegible]MSE[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible]/[illegible]

[illegible],[illegible]:

- 混淆矩阵。
- 准确率/召回率。
- 平均精度
- 精度-召回率曲线。
- ROC曲线和标准。

下面依次讲解,为了让读者更轻松地理解其中的概念,每种指标的例子都会使用良性和恶性乳腺癌。

5.6.1　混淆矩阵

混淆矩阵(confusion matrix)是一种表格,通常用于描述分类模型在一组已知真实值的测试数据上的性能。对于二分类问题混淆矩阵,表中列的标签为预测值,行的标签为实际的标签。在这个乳腺癌的例子中,以下类别为“恶性肿瘤”和“非恶性肿瘤”。

	恶性肿瘤(预测)	非恶性肿瘤(预测)
恶性肿瘤(实际)	23(正确,TP)	1(错误,FN)
非恶性肿瘤(实际)	12(错误,FP)	556(正确,TN)

从上表我们可以看出,在24名患有恶性肿瘤的人中,分类模型正确地预测出23名患者患有恶性肿瘤,这意味着我们有23个真阳性(true positive,TP),即TP=23。而模型预测一个患有恶性肿瘤的人为非恶性肿瘤,这意味着1个假阴性(false negative,FN),即FN=1。同样地,在568名患有非恶性肿瘤,556个被正确地预测,有556个真阴性(true negative,TN),即TN=556。最后,12个非恶性肿瘤被预测为恶性,有12个假阳性(false positive,FP),即FP=12。

[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible]“[illegible]”,[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible],[illegible]

[illegible]:[illegible](precision)[illegible](recall)[illegible]

5.6.2 [illegible]/[illegible]

[illegible]:

$$查准率 \stackrel{\text{def}}{=} \frac{真正}{真正 + 假正}$$

[illegible]:

$$查准率 \stackrel{\text{def}}{=} \frac{真正}{真正 + 假正}$$

[illegible],[illegible]

[illegible],[illegible]([illegible]),[illegible]([illegible])[illegible]

[illegible],[illegible]:

- 算法对数据不敏感的(SVM等模型已经实现了样本加权的功能)。
- 完全随机的,具有相同向量但不同类别特征的数据样本。
- 某类数据样本的样本数过少或非常稀疏的,产生了较为明显的对样本数据的偏好,设置了错误的阈值(在分类器的输出值),而导致某个类别的召回率低于0.9的情况下非常低的召回率问题。

在二分类问题中可以用混淆矩阵分析问题,在多分类中混淆矩阵也可以分析并找出模型分类错误的模式,并可以通过增加样本或修改模型和特征来解决,从而改善分类结果可靠性问题的问题,以提高模型分类的准确率和可靠性。

5.6.3 准确率

准确率(accuracy)是指被正确分类的样本数量占全部被分类的样本数量的比例或者比率,计算公式如下:

$$准确率 \stackrel{\text{def}}{=} \frac{真正 + 真负}{真正 + 真负 + 假正 + 假负} \tag{5.5}$$

当所有类别的预测误差都同等重要时,准确率是一个很有用的评估指标。而对于垃圾邮件/非垃圾邮件的分类问题来说,情况可能并非如此,比如你可能不能容忍假正例的出现而更能接受假负例的出现:假正例意味着你的朋友,发送给你的信息被放入了垃圾邮件文件夹而你看不到,而假负例的问题就不大:一封垃圾邮件进入你的收件箱而你看到了它,并没什么大碍的问题。

5.6.4 代价敏感准确率

当不同类别的样本具有不同的重要性时,代价敏感准确率(cost-sensitive accuracy)是很有用的评估指标。为了计算代价敏感准确率,首先要为两种类型的错误分配代价(一个正数值):假正例和假负例,然后像往常一样

(TP)[illegible](TN)[illegible],[illegible]5.5[illegible]

5.6.5 ROC[illegible]

ROC[illegible]([illegible]“[illegible]”,[illegible])[illegible]

ROC[illegible]([illegible])[illegible]([illegible]),[illegible]

$$\text{真正率} = \frac{\text{正例}}{\text{正例} + \text{假负例}}$$

$$\text{假正率} = \frac{\text{假正例}}{\text{正例} + \text{假负例}}$$

ROC[illegible](confidence)([illegible])[illegible],[illegible]

[illegible]([illegible])[illegible]ROC[illegible]

ROC[illegible]:[illegible][0,1],[illegible]:

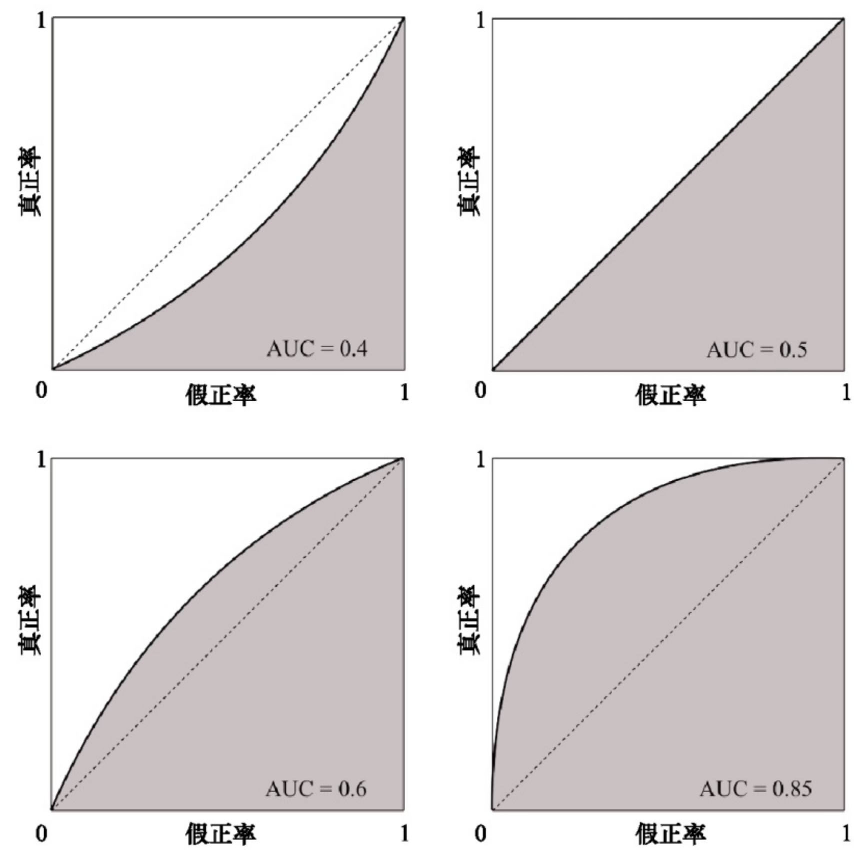
[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1][illegible],[illegible],

[illegible],[illegible]0.7[illegible],[illegible]

[illegible]0.7,[illegible];[illegible],[illegible]

[illegible]5.3[illegible],[illegible]0[illegible],[illegible],[illegible]1[illegible],[illegible]1,

[illegible],[illegible]0,[illegible]



□5.3　ROC□□□□□(□□)

ROC□□□□□(AUC)□□,□□□□□□□□□□AUC□□0.5□□□□□□□□□□□□□□□□AUC□□0.5,□□□□□□□□□□□□□□□□□□□□□□AUC□□□□1□□□□□□,□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1,□□□□□□□□0□

ROC□□□□□□,□□□□□□□□□,□□□□□□□□□(□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□

5.7　□□□□□

[illegible],[illegible],[illegible]ID3[illegible]ϵ[illegible]d[illegible]SVM[illegible]C[illegible]α[illegible]?[illegible]?[illegible]?[illegible],[illegible]

[illegible],[illegible]“[illegible]”[illegible],[illegible],[illegible]

[illegible]([illegible]),[illegible],[illegible](grid search)[illegible]

[illegible]SVM[illegible],[illegible]:[illegible]C([illegible])[illegible][“linear”(“[illegible]”)[illegible]“RBF”][illegible]

[illegible],[illegible]C[illegible],[illegible],[illegible]C[illegible]:[0.001,0.01,0.1,1,10,100,1000][illegible],[illegible]14[illegible]:[(0.001,“linear”),(0.01,“linear”),(0.1,“linear”),(1,“linear”),(10,“linear”),(100,“linear”),(1000,“linear”),(0.001,“rbf”),(0.01,“rbf”),(0.1,“rbf”),(1,“rbf”),(10,“rbf”),(100,“rbf”),(1000,“rbf”)][illegible]

[illegible],[illegible]14[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]

[illegible],[illegible],[illegible]
[illegible],[illegible](random search)[illegible](Bayesian
hyperparameter optimization)[illegible]

[illegible],[illegible],[illegible]
[illegible],[illegible]

[illegible],[illegible],[illegible]
[illegible],[illegible]

[illegible],[illegible](gradient-based technique)[illegible]
(evolutionary optimization technique)[illegible]
[illegible],[illegible],[illegible],
[illegible]

[illegible]

□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□
(cross-validation)□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□(fold)□□□□□□□□5□□□□□□5□□□□□,□□□□□□□□□□5□□:{$F_1,F_2,\ldots,F_5$}□□□$F_k,k=1,\ldots,5$□□□□20%□□□□□□□,□□□□5□□□□□□□□□□f_1□,□□□□F_2,F_3,F_4,F_5□□□□□□□□□□□,□□F_1□□□□□□□□□□□□f_2□,□□□□F_1,F_3,F_4,F_5□□□□□□□□□□□,□□F_2□□□□□□□□□□□□□□□□□□□□□□,□□F_1□F_5□□□□□□□□□□□□□□□□□□5□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□

---

[1] □□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□{□□□□□□}, □□□□□□□□□,□{4,3,2,1}□

[2] □□□□□□□□□k□□□□□□,□□□□□□□□□□□□□□□□□□□□□

[3] □□□□□□□□□□□□□

[illegible]6[illegible] [illegible]

[illegible],[illegible],[illegible],[illegible](logistic regression)![illegible],[illegible]([illegible])[illegible]softmax[illegible],[illegible]

6.1 [illegible]

[illegible],[illegible]

[illegible]SVM[illegible],[illegible](neural network,NN)[illegible]

$$y = f_{NN}(\boldsymbol{x})$$

[illegible]f_{NN}[illegible]:[illegible](nested function)[illegible](layer)[illegible]3[illegible]f_{NN},[illegible]:

$$y = f_{NN}(\boldsymbol{x}) = f_3(\boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{x})))$$

[illegible],$\boldsymbol{f}_1$[illegible]$\boldsymbol{f}_2$[illegible]:

$$f_l(\boldsymbol{z}) \stackrel{\text{def}}{=} \boldsymbol{g}_l(\boldsymbol{W}_l\boldsymbol{z} + \boldsymbol{b}_l) \tag{6.1}$$

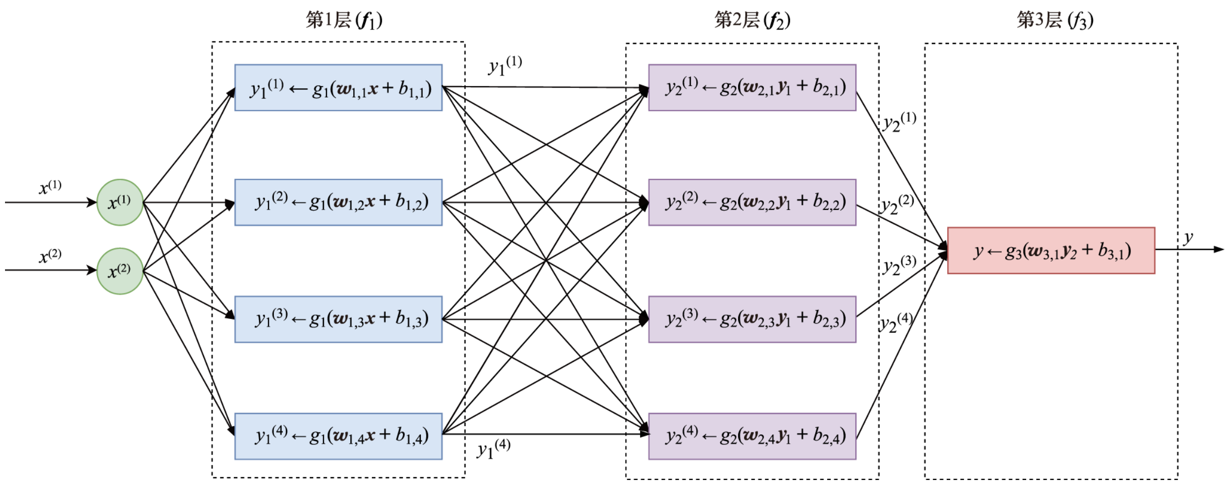
□□, l□□□, □□1□□□□□□□□□□□ $\boldsymbol{g}_l$ □□□□□(activation function)□□□□□□□□□,□□□□□□,□□□□□□□□□□□□□□□□□□□□ $\boldsymbol{W}_l$(□□□□)□ $\boldsymbol{b}_l$(□□□□),□□□□□□□□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□6.1□□□□□□,□□ $\boldsymbol{g}_l$ □□sigmoid□□,□□□□□□ f_3 □□□□□□□□□□□□,□□□□□□□□□□□□□

□□□□□□,□□□□□□□ $\boldsymbol{W}_l$,□□□□□□ $\boldsymbol{w}_l$ □□□□ $\boldsymbol{g}_l$ □□□□□□ $\boldsymbol{W}_l$ □□□□ $\boldsymbol{w}_{l,u}$(u□□□□)□□□ $\boldsymbol{z}$ □□□□□□□ $a_{l,u}=\boldsymbol{w}_{l,u}\boldsymbol{z}+b_{l,u}$, $\boldsymbol{f}_l(\boldsymbol{z})$□□□□□ $[g_l(a_{l,1}),g_l(a_{l,2}),\ldots,g_l(a_{l,\mathrm{size}_l})]$□□□, g_l□□□□□□[1], size_l□ l □□□□□□□□□□□□□,□□□□□□(multilayer perceptron,MLP)□□□□□□□□□□□□□□(vanilla neural network)□

6.1.1 □□□□□□□

□□□□□□□□□□□□MLP□□□□□□□□□,□□□□□□(feedforward neural network,FFNN)□□□□□3□MLP□□,□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□,□□□3□(□□□)□□□□□□

□6.1□□□□MLP□□□□□□6.1□,□□□□□(layer)□□□□□□,□□□□□□□□(□□□,unit)□□□



[illegible]6.1 [illegible]

[illegible]4[illegible]1[illegible]

[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]g,[illegible],[illegible]

[illegible]6.1[illegible],[illegible]g_l[illegible]l,[illegible],[illegible],[illegible]$\boldsymbol{w}_{l,u}$[illegible]$b_{l,u}$,[illegible],l[illegible]u[illegible],[illegible]$\boldsymbol{y}_{l\text{-}1}$[illegible][$y_{l-1}^{(1)}$,$y_{l-1}^{(2)}$,$y_{l-1}^{(3)}$,$y_{l-1}^{(4)}$][illegible]$\boldsymbol{x}$[illegible][$x^{(1)}$,...,$x^{(D)}$][illegible]

[illegible]6.1[illegible],[illegible](fully-connected layer)[illegible]

6.1.2 [illegible]

最后一层可用的激活函数由特定任务类型确定，如果是回归，则最后一层(输出层)包含单个单元的隐层，激活函数是线性的，如果g_{last}是线性的，如果是二分类，则最后一层包含单个单元，g_{last}是逻辑回归函数，以此类推各种类型的问题。

对于隐层的其他层，神经网络函数[2]单元函数$g_{l,u}$可以写成如下形式，向量l中u单元$\boldsymbol{w}_{l,u}$和$b_{l,u}$是参数，而每一层中的单元会形成不同函数的连接，即f_{NN}中的函数会形成一个嵌套函数，对于每一个嵌套函数，我们会使用一个矩阵和一个向量作为参数，而线性函数会发生两次，f_{NN}中的所有函数都是线性的，$\boldsymbol{W}_l z+\boldsymbol{b}_l$也是一个线性函数，所以需要非线性激活函数才可以表示非线性关系。

激活函数的选择有很多种，有些是逻辑回归函数的变体，最常用的TanH和ReLU。前者是双曲正切函数，与逻辑函数类似，但取值范围在-1到1(不包括-1和1)之间，后者是修正线性单元函数，当输入z小于零时，输出0，否则输出z本身：

$$tanh\left(z\right)=\frac{\mathrm{e}^{z}-\mathrm{e}^{-z}}{\mathrm{e}^{z}+\mathrm{e}^{-z}},$$

$$relu(z)=\begin{cases}0 & z<0\\ z & \text{其他}\end{cases}$$

如上所述，$\boldsymbol{W}_l z+\boldsymbol{b}_l$中的$\boldsymbol{W}_l$是一个矩阵，$\boldsymbol{b}_l$是一个向量，和线性回归中的表达式$\boldsymbol{w}z+b$类似。矩阵$\boldsymbol{W}_l$中，每一行$u$对应一个参数向量$\boldsymbol{w}_{l,u}$，向量$\boldsymbol{w}_{l,u}$的维度等于第$l$-1层的单元数，$\boldsymbol{W}_l z$运算会输出一个向量$\boldsymbol{a}_l \stackrel{\text{def}}{=}[\boldsymbol{w}_{l,1}z,\boldsymbol{w}_{l,2}z,\cdots,\boldsymbol{w}_{l,\text{size}_l}z]$，然后，$\boldsymbol{a}_l+\boldsymbol{b}_l$会输出一个长度为$\text{size}_l$的向量$\boldsymbol{c}_l$，最后，$g_l(\boldsymbol{c}_l)$会输出一个向量$\boldsymbol{y}_l \stackrel{\text{def}}{=}[y_l^{(1)},y_l^{(2)},\cdots,y_l^{(\text{size}_l)}]$。

6.2 深度学习

深度学习指的是训练多层神经网络，在过去的多层神经网络训练中，随着网络层数的增加，训练变得越来越难，其中最大的两个挑战是梯度爆炸和梯度消失问题，在使用梯度下降法训练神经网络时，梯度爆炸(exploding

gradient)□□□□□(vanishing gradient)□

□□□□,□□□□□□□□□□□□□□□□□□□□□□(gradient clipping)□□L1□L2□□□□□□□□□□□□□□□□□□□□□□□

□□,□□□□□□□?□□□□□□□□□□?□□□□□,□□□□□□□□□□(backpropagation)□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□4□□,□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□(0,1)□□□□□□□□□□□□□□□□□□□*n*□□□□,□□□□□(□□□)□□□□□□*n*□□□□□□□□□□□□,□□□□*n*□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□)□□□□□□□□□□□□□□,□□ReLU□LSTM(□□□□□□□□,□□□□□□□)□□□□□□□□□□□□□(residual neural network)□□□□□□(skip connection),□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□)□“□□□□”□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□(hidden layer)□

6.2.1 □□□□□

[illegible],MLP[illegible],[illegible]$(\text{size}_l+1)\cdot\text{size}_l$[illegible]([illegible]$\boldsymbol{W}_l$[illegible]$\boldsymbol{b}_l$)[illegible],[illegible]1 000[illegible],[illegible]100[illegible](computationally intensive)[illegible]

[illegible],[illegible][3][illegible]MLP[illegible],[illegible](intractable)[illegible]

[illegible](Convolutional Neural Network,CNN)[illegible]FFNN[illegible],[illegible]CNN[illegible],[illegible]CNN[illegible],[illegible]

[illegible]CNN[illegible],[illegible]

[illegible],[illegible]:[illegible]“[illegible]”[illegible]:[illegible]“[illegible]”[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible](patch)[4][illegible],[illegible],[illegible],[illegible](pattern)[illegible],[illegible],[illegible],[illegible]

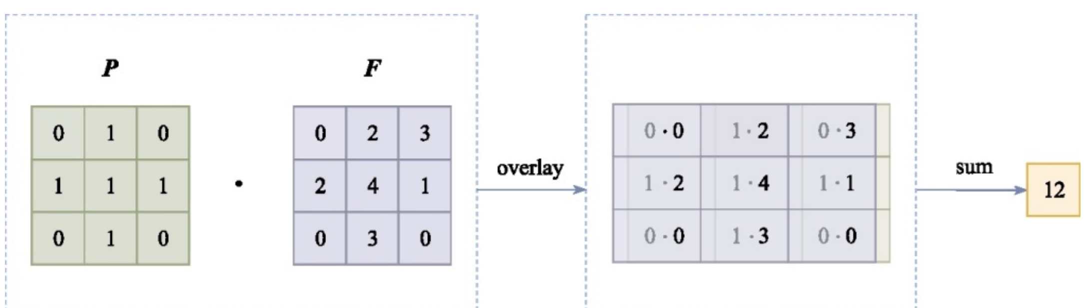
□CNN□,□□□□□□□□□□□6.1□□□,□□□□□□1□□□□□2□3□□□□□□□□□□,□□□□□□□□□□□□□□$p\times p$□□□$\boldsymbol{F}$(□□□,filter□□□□□)□□□□□□,p□□□□□□□□□□□□□□□□□□□□□□,1□0□□□□□□□□□□□□□□□□□□□□□□3×3□□(p=3)□□□□□□□□□□□□□□□□$\boldsymbol{P}$(□□,patch□□□□□):

$$\boldsymbol{P}=\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}$$

□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□)□□□□□□□□□□□□□3×3□□□□□$\boldsymbol{F}$□□□□□□□□1□□□□□□□□□,□□0□□□□□□□□□□□□□□□□□$\boldsymbol{P}$□$\boldsymbol{F}$□□□(convolution),$\boldsymbol{F}$□$\boldsymbol{P}$□□□□□□□□□□□□□□□□□□□□□,□□□□$\boldsymbol{F}$□:

$$\boldsymbol{F}=\begin{bmatrix}0&2&3\\2&4&1\\0&3&0\end{bmatrix}$$

□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{P}$□$\boldsymbol{F}$□□□□□□□□□6.2□□□



□6.2 □□□□□□□□□□

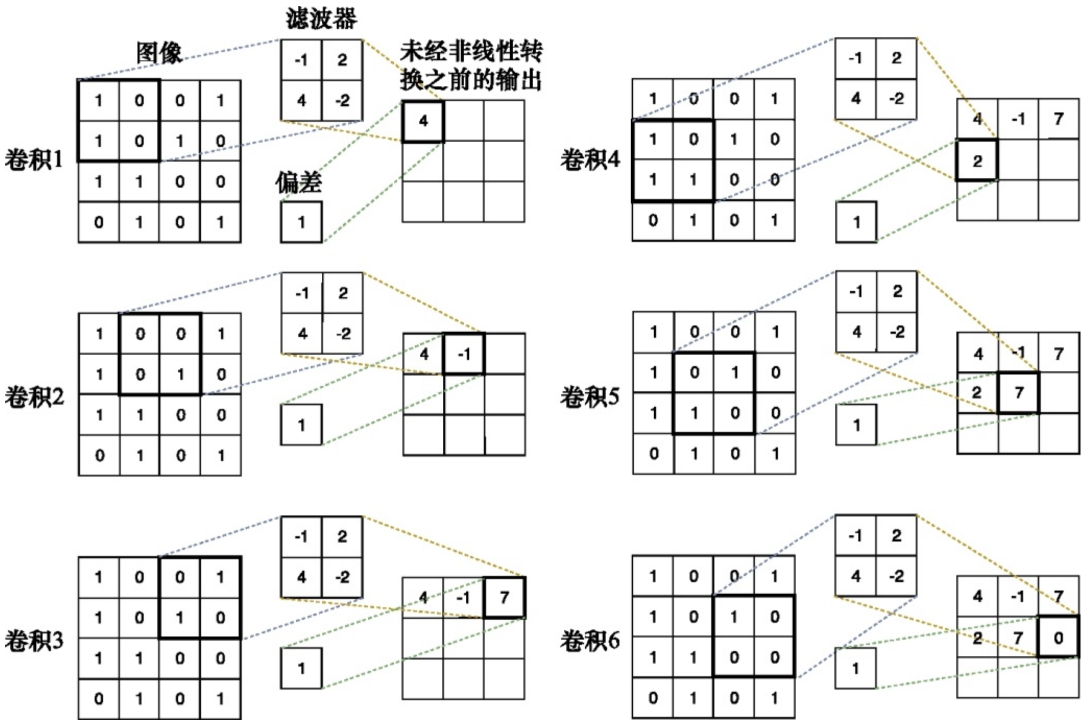
□□□□□□□$\boldsymbol{P}$□□□□□□□□□,□□□□□□□L□:

$$\boldsymbol{P}=\begin{bmatrix}1&0&0\\1&0&0\\1&1&1\end{bmatrix}$$

□□,□□***F***□□□□□□□□□:5□□□□□,□□□□□□□□□□“□□□”□□□,□□□□□□□□□□□□□□□,□□□□□□□□***F***□□□□□□□□*b*,□□□□□□(□□□□)□□□□□□□□□□

□□CNN□□□□□□□□□□□(□□□□□□□□□□),□□□□□□FFNN□□□□□□□□□(□□)□□□□□□□,□□□□,□□□□,□□□□□□□□□□,□□□□□□□□□□□□□

□6.3□6□□□□□□□□□□□□□□□□□□□□□



□6.3 □□□□□□□□□□□□□□

□□□□□(□□□□□□□)□□□□□□□□□□□□,□□□□□□□□□□□□□□□

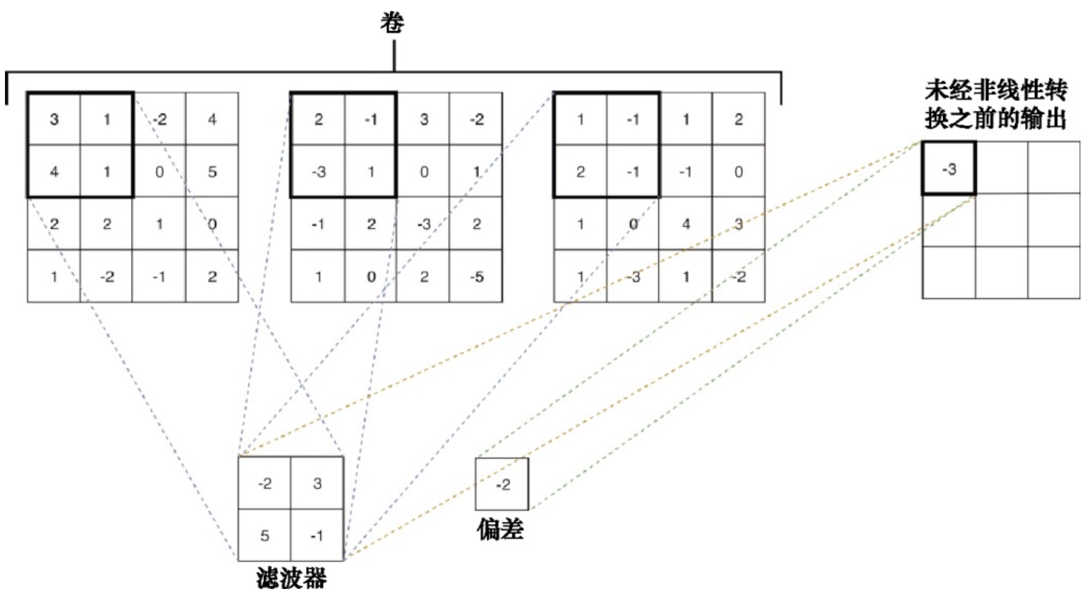
□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□ReLU□□□□□□□□□□□□□□□□□□□□□□□

□□□□□*l*□□$size_l$□□□□,□□□□□□□□□□□$size_l$□□□,□□□□□□□□□□

□□CNN□□□□□□□□□□□□,□/+1□□□/□□□□□□size/□□□□□□□□□□□□□□□□□□□□□(volume),□□□□□□□□□□□□/+1□□

□□,□□□□□□□3□□□□□□□□□□□□□6.4□□□□□□□□□□□□:

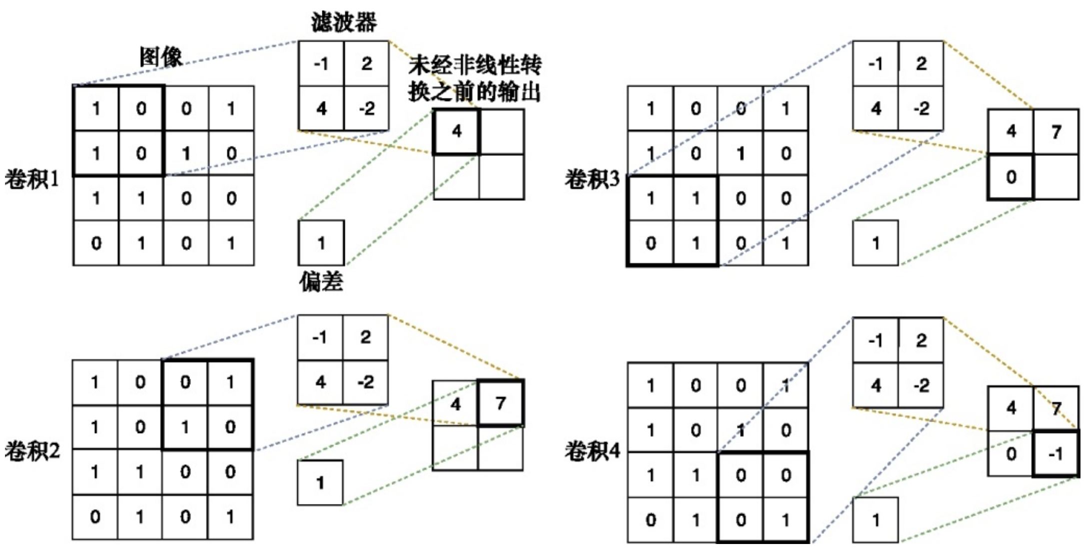
[-2·3+3·1+5·4+(-1)·1]+[(-2)·2+3·(-1)+5·(-3)+(-1)·1]+[(-2)·1+3·(-1)+5·2+(-1)·(-1)]+(-2)=-3



□6.4　□□□□□□□3□□□

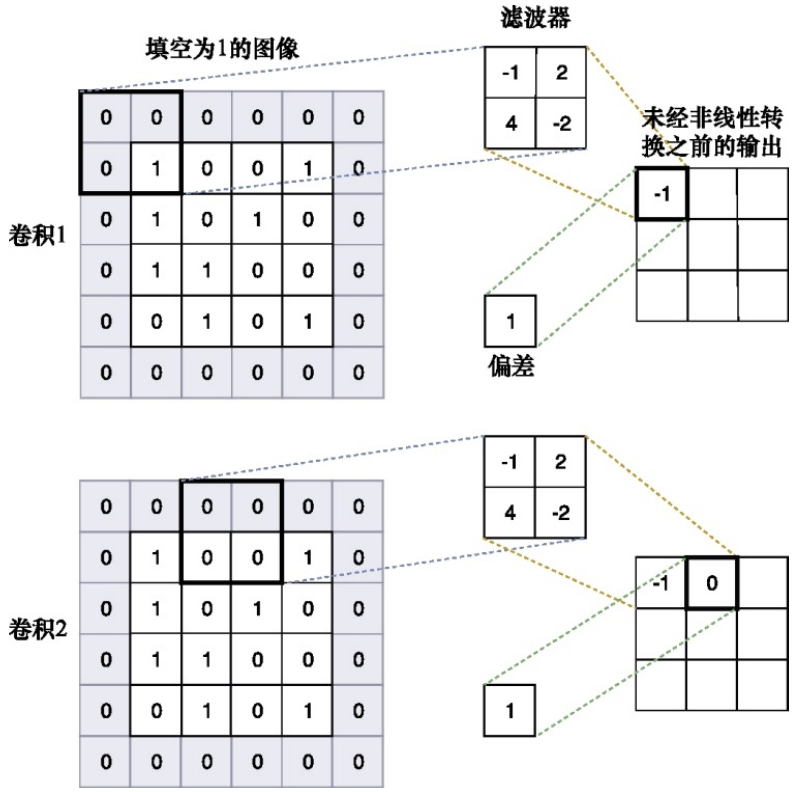
□□□□□□□□,□□□□3□□□(R□G□B)□□,□□□□□□□□□□□□□□□□,CNN□□□□□□□□

□□□□□□□□□□□□□(stride)□□□(padding)□□□6.3□□□□,□□□□□1,□□□□□□□□□□□□□□□□□□□□(cell)□□□6.5□,□□□□□□□□□□□□□□2□□□□□□□□□□□□□□,□□□□□□,□□□□□□□□□□



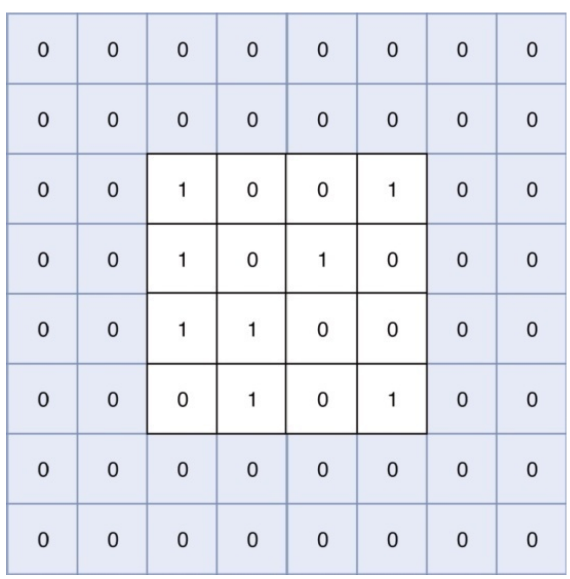
□6.5　□□□2□□□

□□□□,□□□□□□□;□□□□□□□□□□□□□□□□□,□□□□□(□□)□□□□□□□□□□□□□□□□□□□□□□0□□□6.3□,□□□0,□□□□□□□□□□□□□□□□□□□6.6□,□□□2,□□□1,□□□□□□□□□□□□□□1□□□□□□□□,□□□□□□□,□□□□□□□[5]□



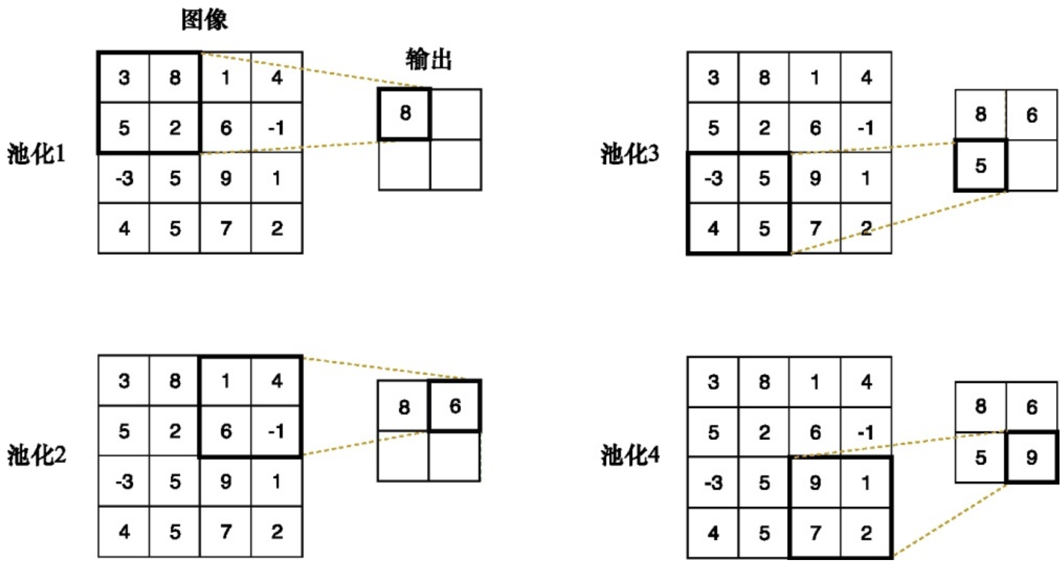
[illegible]6.6 [illegible]2[illegible]1[illegible]

[illegible]6.7[illegible]2[illegible]“[illegible]”[illegible]



[illegible]6.7 [illegible]2[illegible]

□□,□□□□□□□□□□(pooling),□□□□□□CNN□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□(max)□□□□(average)□□□□□□□□□□,□□□□□□□:□□□□□□□□□□□□6.8□□,□□□□□□□□□□□□□□□□□2□



□6.8　□□□□□2□□□□2□□□□□

□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□2□3,□□□□2□□□□□□□□,□□□□□□□,□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□(□□6.8□□,□□□□□□□□□□□2□,□□□□□□□)□

6.2.2　□□□□□□

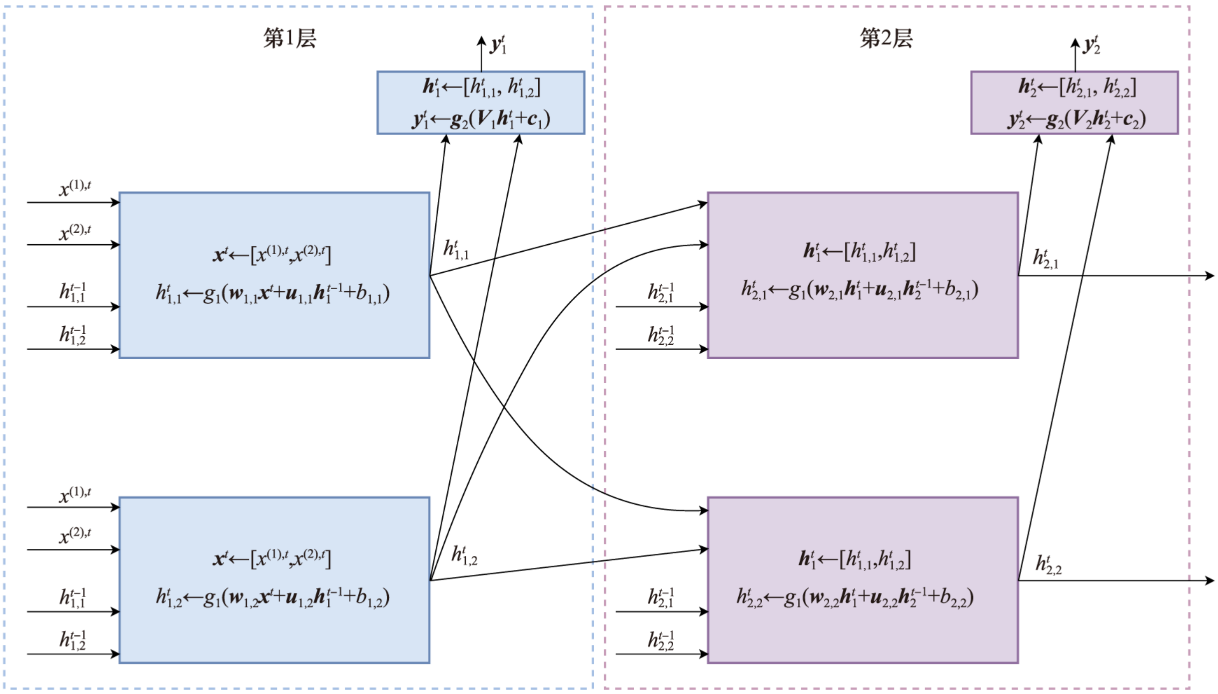
□□□□□□(Recurrent Neural Network,RNN)□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□(□□□□□□□)□

RNN□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□□□□□□,□□□l□□□□□u□□□□□□□□(state)□□□□$h_{l,u}$□□□□□□□□□□□□□□RNN□,□□□l□□□□□□u□□□□□□:□□□l-1□□□□□□□,□□□□□l□□□□□□□□□□□□

□□□□□□□□,□□□□□RNN□□□□□□□□□□□□□(□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□6.9□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□$\boldsymbol{X}=[\boldsymbol{x}^1,\boldsymbol{x}^2,\ldots,\boldsymbol{x}^{t-1},\boldsymbol{x}^t,\boldsymbol{x}^{t+1},\ldots,x^{\mathrm{length}_x}]$,$\mathrm{length}_{\boldsymbol{x}}$□□□□□□□□□□□□□□□□□□$\boldsymbol{X}$□□□□□,□□□□□□□$\boldsymbol{x}^t$□□□□□□□$t$□□□,$t$=1,…,$\mathrm{length}_{\boldsymbol{x}}$□



□6.9 □□RNN□□□□

□□□□□□□□□□□□□□□□□

□□6.9□□,□RNN□,□□□□□□□□□□□□□□□□(timestep)□□“□□”□□□t□□□□□□□□□□□□□□□□□t,□□□l□□□□u□□□$h_{l,u}^{t}$□□□,□□□□□□□□□□□□$\boldsymbol{h}_{l,u}^{t-1}$□□□□□$\boldsymbol{h}_{l,u}^{t-1}$□□□□□□□□□$t$-1□□□□□□□□□□□□□□□□□□$\boldsymbol{w}_{l,u}$,$\boldsymbol{u}_{l,u}$□□□□□$b_{l,u}$□□□□□□,□□□□□□□□□□□□□$g_1$,□□$\boldsymbol{h}_{l,u}^{t}$□□□□□□,□□□$tanh$□□□□□$g_1$□□$\boldsymbol{y}_l^t$□□□□□□,□□□□□□$l$□□□□□$\boldsymbol{y}_l^t$,□□□□□□□□□$g_2$□□□□□□,$g_2$□□□□□□□□□□□□□□□,□□□□□□□$\boldsymbol{h}_{l,u}^{t}$□□□□□□$g_2$□$\boldsymbol{h}_{l,u}^{t}$□□□□□□$\boldsymbol{V}_l$□□□□□□$\boldsymbol{c}_{l,u}$□□□□□□□□□,□□□**softmax**□□□g_2:

$$\boldsymbol{\sigma}(z) \stackrel{\text{def}}{=} [\sigma^{(1)}, \cdots, \sigma^{(D)}], \text{□□} \quad \boldsymbol{\sigma}^{(j)} \stackrel{\text{def}}{=} \frac{\exp\left(z^{(j)}\right)}{\sum_{k=1}^{D} \exp\left(z^{(k)}\right)}$$

softmax□□□sigmoid□□□□□□□□□□□□□□□□□□□□□:$\sum_{j=1}^{D}\sigma^{(j)}=1$,□□□□$j$□$\sigma^{(j)}>0$□

□□$\boldsymbol{V}_l$□□□□□□□□□□□□□$\boldsymbol{V}_l$□□□$\boldsymbol{h}_l^t$□□□□□□□□□□$\boldsymbol{c}_l$□□□□□□□□□□□□□□□□□□□□□□□□□□y□□□□(□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□)

□□□□□□□□□□$\boldsymbol{w}_{l,u}$,$\boldsymbol{u}_{l,u}$,$b_{l,u}$,$\boldsymbol{V}_{l,u}$□$\boldsymbol{c}_{l,u}$□□,□□□□□□□□□□□□□□□□RNN□□□□□□□□□□□□□——□□□□□□□□(backpropagation through time)□

tanh□softmax□□□□□□□□□□□□□□,□□□□□□□□□□□□□RNN□□□□□□□□□□,□□□□□□□□□□□□“□□”□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□

RNN□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□RNN(gated RNN)□□□□□□□□□□□□(long short-term memory,LSTM)□□,□□□□□□□□□□□(gated recurrent unit,GRU)□□□□

□RNN□□□□□□(gated unit)□□□□,□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□(bit)□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□(0,1)□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□,□□□□□□□□t□□□□□□□□□□

□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□RNN□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□(gate)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□GRU(minimal gated GRU),□□□□□□□□□□□□□□□□

□□□RNN□□□□(□□□□□□□□□□□□□)□□□□□□□GRU□□□□□□□□□□□□□□□□□,□l□□□□□□□□GRU□□u□□□□□□:□□□□□□□□□□□□□□$\boldsymbol{h}_l^{t-1}$□□□□□□□□$\boldsymbol{x}^t$□□□□,□□□□□□□□□□□□□(□□□□□□□□□□□□□□□)□

$$
\begin{aligned}
\widetilde{h}_{l,u}^t &\leftarrow g_1(\boldsymbol{w}_{l,u}\boldsymbol{x}^t + \boldsymbol{u}_{l,u}\boldsymbol{h}_l^{t-1} + b_{l,u}), \\
\Gamma_{l,u}^t &\leftarrow g_2(\boldsymbol{m}_{l,u}\boldsymbol{x}^t + \boldsymbol{o}_{l,u}\boldsymbol{h}^{t-1} + a_{l,u}), \\
h_{l,u}^t &\leftarrow \Gamma_{l,u}^t\, \widetilde{h}_l^t + (1-\Gamma_{l,u}^t)h_l^{t-1}, \\
\boldsymbol{h}_l^t &\leftarrow [h_{l,1}^t, \cdots, h_{l,\text{size}_l}^t] \\
\boldsymbol{y}_l^t &\leftarrow g_3(\boldsymbol{V}_l\boldsymbol{h}_l^t + \boldsymbol{c}_{l,u})
\end{aligned}
$$

□□,g_1□□□tanh□□□□,g_2□□□□□(gate function),□sigmoid□□□□□(0,1)□□□$\Gamma_{l,u}$□□□□0□,□□□□□□□□□□□□□□h_l^{t-1}□□□□□□□,□□$\Gamma_{l,u}$□□□□1□,□□□□□□□□□□□□$\widetilde{h}_{l,u}^t$□□(□□□□□□3□□□)□□□□RNN□□,g_3□□□softmax□□□

[illegible](identity function, $f(x)=x$)[illegible],[illegible]

[illegible]RNN[illegible](bi-directional RNN),[illegible](attention)[illegible](sequence-to-sequence RNN)[illegible],[illegible]RNN[illegible](RecursiveNeuralNetwork)[illegible]

---

[1] [illegible],[illegible],[illegible]

[2] [illegible],ReLU[illegible]0[illegible]

[3] [illegible]100×100[illegible],[illegible]10 000[illegible]

[4] [illegible],[illegible],[illegible]

[5] [illegible],[illegible]6.6[illegible]9[illegible]

[illegible]7[illegible] [illegible]

7.1 [illegible]

[illegible],[illegible],[illegible],[illegible]?[illegible] $\{(x_i, y_i)\}_{i=1}^{N}$,[illegible](mean squared error,MSE),[illegible],[illegible] w_1,w_2[illegible]b[illegible],[illegible],[illegible]D[illegible](D>3)[illegible],[illegible]

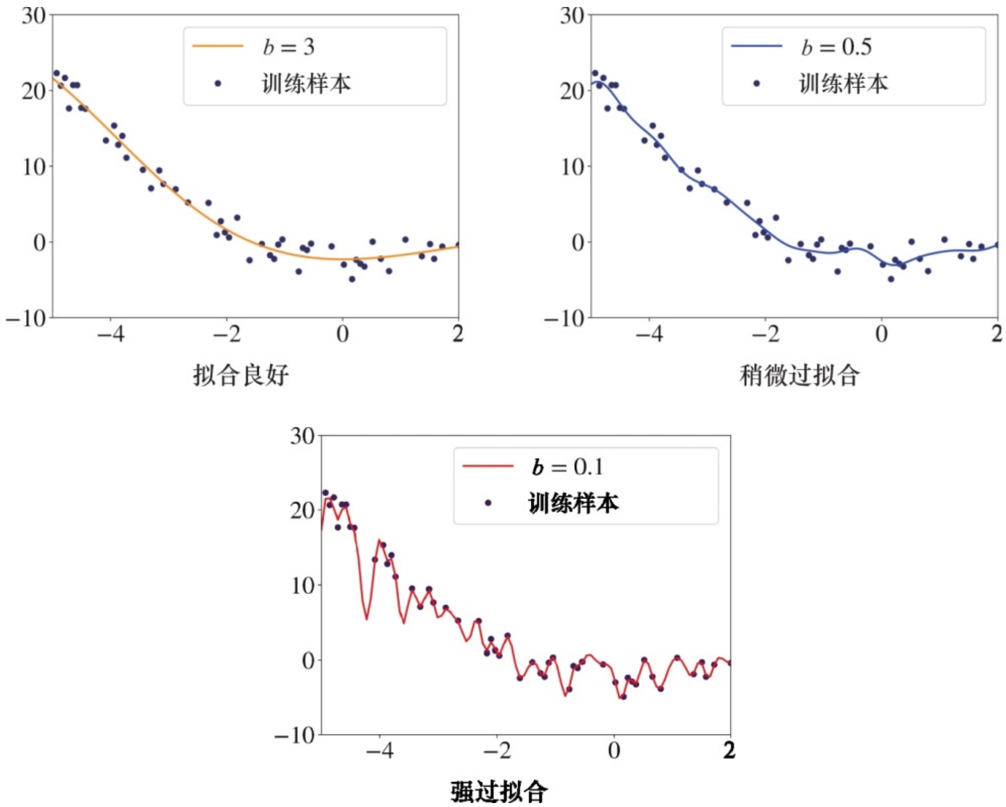
[illegible](non-parametric method)[illegible],[illegible],[illegible]([illegible]kNN[illegible])[illegible],[illegible],[illegible]:

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} w_i y_i, \quad \text{[illegible]} \quad w_i = \frac{N k\left(\frac{x_i - x}{b}\right)}{\sum_{l=1}^{N} k\left(\frac{x_l - x}{b}\right)} \tag{7.1}$$

[illegible]$k(\bullet)$[illegible](kernel)[illegible]:x[illegible]x_i[illegible],[illegible]w_i[illegible];[illegible],[illegible](Gaussian)[illegible]:

$$k(z) = \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)$$

□7.1□,*b*□□□□□□,□□□□□□□□□□□□□(□□□□□*b*□□□□□,□□□□□□□□□□MSE)□□□□7.1,□□□□*b*□□□□□□□□□□□



□7.1　□□□□□□3□□□*b*□□□□□□

□□□□□□□□□□□□□□,□7.1□□x_i-x□x_l-x□□□□□□□□□□□||$\boldsymbol{x}_i$-$\boldsymbol{x}$||□||$\boldsymbol{x}_l$-$\boldsymbol{x}$||□

7.2　□□□□□

□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□C□□□□□□:y∈{1,...,C}□□□□□□□□□□□□□,□□□□□□□SVM□□□□□□□□□□□□□□□□ID3□□□□□□□□□□□□□□□□□□□:

$$f_{\mathrm{ID3}}^{S} \stackrel{\mathrm{def}}{=} \Pr(y_i = c|\boldsymbol{x}) = \frac{1}{|S|} \sum_{\{y|(\boldsymbol{x},y)\in S, y=c\}} y$$

□□,c□□□c∈{1,...,C},S□□□□□□□□□

□□□□□□□□,□□□sigmoid□□□**softmax**□□,□□□□□□□□□□□□□6□□,□□□□□□softmax□□□

kNN□□□□□□□□□□□□□□:□□□□□□□$\boldsymbol{x}$□□□k□□□,□□□□□□□□□□,□□□□□□□□□□□□□

SVM□□□,□□□□□□?□□□□□□□□□□□(one versus rest)□□□□□□,□□□□□□□□□□□C□□□□□□□,□□□C□□□□□□□□□□□,□□□□□□□3□□□y∈{1,2,3},□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□1□□□□□□□□0□□□□□□□□,□□□□□□□□2□□□□□□□□0□□□□□□□□,□□□□□□□3□□□□□□□□0□□□□□,□□□□□3□□□□□□□,□□□□□□1□0□2□0□3□0□

□□3□□□□□,□□□□□□□□□$\boldsymbol{x}$□□□□□,□□□□□□□□,□□□3□□□□□□,□□□□□□□(the most certain)□□0□□□□□□□□□□□□□□□,□□□□□□□□□□□□(0□1□□)□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□(certainty)□□SVM□,□□□□□□□□$\boldsymbol{x}$□□□□□□□□□d,□□□□□□:

$$d \stackrel{\text{def}}{=} \frac{\boldsymbol{w}^{*}\boldsymbol{x} + b^{*}}{||\boldsymbol{w}||}$$

[illegible],[illegible],[illegible]

7.3 [illegible]

[illegible],[illegible],[illegible]

[illegible](one-class classification)[illegible](unary classification)[illegible](class modeling)[illegible], [illegible],[illegible],[illegible],[illegible],[illegible],[illegible](outlier detection,anomaly detection)[illegible](novelty detection)[illegible]

[illegible],[illegible](one-class Gaussian)[illegible]**k**[illegible]**kNN**[illegible]**SVM**[illegible]

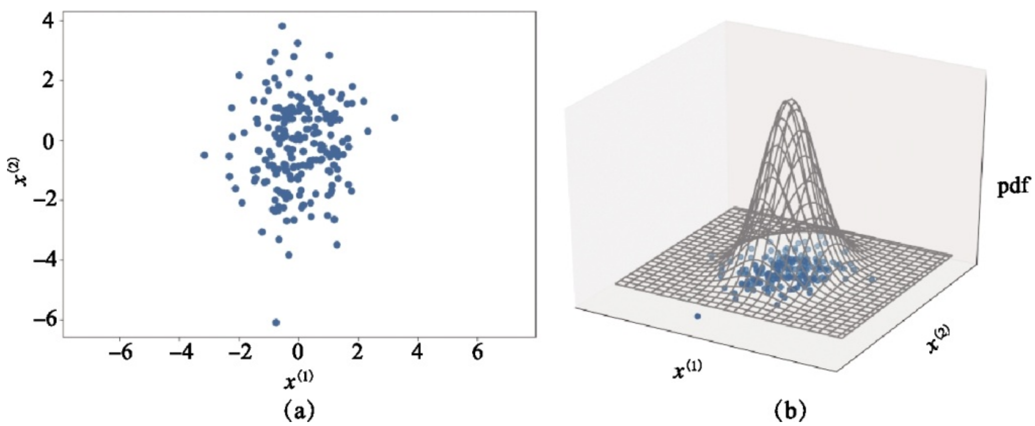
[illegible],[illegible],[illegible](multivariate normal distribution,MND),[illegible]MND[illegible](probability density function,pdf)[illegible]:

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^{D}|\boldsymbol{\Sigma}|}}$$

□□,$f_{\mu,\Sigma}(\boldsymbol{x})$□□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□,□□□MND□□□□□□$\boldsymbol{\mu}$(□□□□)□$\boldsymbol{\Sigma}$(□□□□)□□□□□□□□□□□□□□□□□(maximum likelihood)□□(□□□□□□□□□□□□)□□□□□□□□□□$|\boldsymbol{\Sigma}| \overset{\text{def}}{=} \det\boldsymbol{\Sigma}$,□□□$\boldsymbol{\Sigma}$□□□□(determinant);$\boldsymbol{\Sigma}^{-1}$□□$\boldsymbol{\Sigma}$□□□□(inverse)□

□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□(matrix theory)□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

□□□□,□□$\boldsymbol{\mu}$□□□□□□□□□□□□□□□□□□□□$\boldsymbol{\Sigma}$□□□□□□□□□□□□7.2□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□



□7.2 □□□□□□□□□□□□□□□□□

□:(a)□□□□□;(b)□□7.2(a)□□□□□□□MND□□□

□□□□□□□□□□□$\boldsymbol{\mu}$□$\boldsymbol{\Sigma}$□,□□□□$f_{\mu,\Sigma}(\boldsymbol{x})$□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□□□□,□□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□□,□□□“□□□□□(educated guess)”□□□

□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□(□□□□)□□□,□□□□□□□□□□□□:□□□□□□□1□**μ**□1□**Σ**,□□□□□□□□□□□□□□□□□□□□□□□□□□□□9□□,□□□□□□□□□□□□□□□□□□□□□

□□k□□□□□kNN□□□□□□□□□□□□□□:□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□k□□□,□□□□□□□□**k**□□(k-mean)□□□□□□□□□□**x**,□□*d*(**x**)□□**x**□□□□□□□□□,□□□□□□□□*d*(**x**)□□□□□□,□**x**□□□□□□

□□SVM,□□□□□□□,□□□□□(□□□□□□)□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□(hypersphere)□□□□□□□□□□kNN□□,□□□□k□□□□□SVM□□□□□,□□□□□□□□□□□□□

7.4 □□□□□

□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□(multi-label classification)□

□□,□□□7.3□□□□,□□□□□□□□□□□:“□”“□”“□□”□



[illegible]7.3 [illegible]"[illegible]""[illegible]"[illegible]"[illegible]"[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]:[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]([illegible])[illegible],[illegible]

[illegible](binary cross-entropy)[illegible],[illegible],[illegible],[illegible]sigmoid[illegible],[illegible]/[illegible]

$(y_{i,l} \in \{0,1\})$,[illegible]$l=1,\ldots,L$[illegible]$i=1,\ldots,N$[illegible]$\boldsymbol{x}$[illegible]l[illegible]$\hat{y}_{i,l}$,[illegible]

[illegible]:

$$-\left(y_{i,l} \ln\left(\hat{y}_{i,l}\right)+\left(1-y_{i,l}\right) \ln\left(1-\hat{y}_{i,l}\right)\right)$$

[illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible],[illegible]{[illegible]};[illegible]{[illegible]}[illegible],[illegible]:

[illegible]	**[illegible]1**	**[illegible]2**
1	[illegible]	[illegible]
2	[illegible]	[illegible]
3	[illegible]	[illegible]
4	[illegible]	[illegible]
5	[illegible]	[illegible]
6	[illegible]	[illegible]

[illegible]1[illegible]6[illegible]

[illegible]

7.5 [illegible]

[illegible]3[illegible](ensemble learning)[illegible]

[illegible](meta-model)[illegible]

[illegible](weak learner)[illegible]

[illegible]*x*[illegible]*x*[illegible]

[illegible](boosting)[illegible](bagging)[illegible]

7.5.1 [illegible]

[illegible],[illegible]“[illegible]”[illegible],[illegible](ensemble model)[illegible]

[illegible]“[illegible]”([illegible])[illegible],[illegible],[illegible](random forest)[illegible],[illegible]

7.5.2 [illegible]

[illegible],[illegible]B[illegible]S_b(b=1,...,B),[illegible]S_b[illegible]f_b[illegible]S_b[illegible],[illegible](sampling with replacement)[illegible],[illegible],[illegible]S_b,[illegible],[illegible]$|S_b|=N$[illegible]

[illegible],[illegible]B[illegible],[illegible]$\boldsymbol{x}$[illegible]B[illegible]:

$$y \leftarrow \hat{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{B}\sum_{b=1}^{B} f_b(\boldsymbol{x})$$

[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible]

样本进行一种"平均"的方式来获得一个结果,在每一种情况下获得的不同结果会因为其中相同的样本或部分相同的样本而形成偏差,因此能够形成一个更加稳定的模型,同时能够降低在小样本数据上出现的误差对所有结果产生影响,这是随机森林的一个显著优势。

随机森林中的最佳超参数是树的数量B以及每个分裂所考虑的特征数量。

为什么随机森林比基于原始数据集的单一决策树表现更好?其原因,通过使用许多原始的样本,我们降低了最终模型的方差(variance),而方差是导致过拟合(overfitting)的主要问题。过拟合产生的原因在于,模型会试图去解释数据集中的少量偏差的样本,因为数据集只是所有可能的示例中的一个小样本集合,为了去解释它的噪声和异常,而模型在训练集中的这些噪声上的表现(如出现在训练集中)会出现问题:在没有噪声时,噪声会产生影响使得模型对一些不存在的样本进行预测,从而降低了模型在其他样本上的效果。

7.5.3 梯度提升

梯度提升法(gradient boosting)是另一种有效的集成学习算法。先看回归问题中的梯度提升。我们从一个常数模型开始构建一个强回归模型,首先初始化一个常数模型$f=f_0$开始(例如ID3算法中):

$$f=f_0(\boldsymbol{x})\stackrel{\text{def}}{=}\frac{1}{N}\sum_{i=1}^{N}y_i$$

然后,修改训练集中的每个样本$i=1,\ldots,N$的标签为如下形式:

$$\hat{y}_i\leftarrow y_i-f(\boldsymbol{x}_i)$$

(7.2)

其中,$\hat{y}_i$称为残差(residual),是样本$\boldsymbol{x}$的新标签。

[illegible],[illegible]f_1[illegible],[illegible] $f \stackrel{\text{def}}{=} f_0 + \alpha f_1$,[illegible]$\alpha$[illegible]([illegible])[illegible]

[illegible],[illegible]7.2[illegible],[illegible],[illegible]f_2,[illegible]$f \stackrel{\text{def}}{=} f_0 + \alpha f_1 + \alpha f_2$[illegible],[illegible]$M$[illegible]

[illegible]?[illegible],[illegible]f[illegible]([illegible])[illegible],[illegible]([illegible]),[illegible]α[illegible],[illegible],[illegible]M([illegible])[illegible]

[illegible]?[illegible]4[illegible],[illegible],[illegible],[illegible]:[illegible]MSE[illegible],[illegible],[illegible]α,[illegible],[illegible]:[illegible],[illegible]([illegible])[illegible]

[illegible]3[illegible]3[illegible]:[illegible],[illegible]

[illegible]f[illegible]:[illegible]([illegible])[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]M[illegible]sigmoid[illegible],[illegible]:

$$\Pr(y=1|\boldsymbol{x},f)\overset{\text{def}}{=}\frac{1}{1+\mathrm{e}^{-f(\boldsymbol{x})}}$$

□□, $f(\boldsymbol{x})\overset{\text{def}}{=}\sum_{m=1}^{M}f_m(\boldsymbol{x})$, f_m□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□ $L_f=\sum_{i=1}^{N}\ln[\Pr(y_i=1|\boldsymbol{x}_i,f)]$ □f□□□,□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□

□□,□□□□□□□□□□ $f=f_0=\frac{p}{1-p}$,□□ $p=\frac{1}{N}\sum_{i=1}^{N}y_i$ (□□□□□□□□□□sigmoid□□□□□□)□□□,□□□□□m□,□□□□f_m□□□□□□□□□□□□f_m,□□□□i=1,…,N□□□□□□□□□g_i:

$$g_i=\frac{\mathrm{d}L_f}{\mathrm{d}f}$$

□□,f□□□□□□□m-1□□□□□□□□□□□□□g_i,□□□□□ln[Pr(y_i=1|$\boldsymbol{x}_i$,f)]□f□□□□i□□□□□□□□□□,□ln[Pr(y_i=1|$\boldsymbol{x}_i$,f)] $\overset{\text{def}}{=}\ln\left[\frac{1}{1+\mathrm{e}^{-f(\boldsymbol{x}_i)}}\right]$□,□□□□□$f$□□□□□ $\frac{1}{\mathrm{e}^{f(\boldsymbol{x}_i)}+1}$□

□□,□□□□□□□,□□□□y_i□□□□□□□□□g_i,□□□□□□□□□□□□□f_m□□□□□□□□□□□□□□ρ_m:

$$\rho_m\leftarrow\arg\max_{\rho}L_{f+\rho f_m}$$

□□□m□□□,□□□□□□□□f,□□□□f_m:

$$f \leftarrow f + \alpha \rho_m f_m$$

□$m=M$□,□□□□,□□□□□□□f□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□

7.6 □□□□□□

□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□;□□□□□□□□□;□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□

□□□□(sequence labeling)□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□($\boldsymbol{X}$,$\boldsymbol{Y}$)□□□,$\boldsymbol{X}$□□□□□□□,□□□□□□□□□□(timestep);$\boldsymbol{Y}$□□□□□□□□□□□□,$\boldsymbol{X}$□□□□□□□□□□□,□[“□”“□□”“□□”],$\boldsymbol{Y}$□□□□□□□□□(part of speech),□[“□□□”“□□□”“□□”]□□□□□□□□□,□□□□□i□,$\boldsymbol{X}_i$=[$\boldsymbol{x}_i^1$,$\boldsymbol{x}_i^2$,…,$\boldsymbol{x}_i^{\mathrm{size}_i}$]□□□,size$_i$□□□$i$□□□□□,$\boldsymbol{Y}_i$=[$y_i^1$,$y_i^2$,…,$y_i^{\mathrm{size}_i}$]□$y_i$∈{1,2,…,$C$}□

□□□RNN□□□□□□□,□□□□□□□□□□□□□□□□□□□□□t,RNN□□□□□□□□□$\boldsymbol{x}_i^{(t)}$,□□□□□□□□□□□□□□□$\boldsymbol{y}_{\mathrm{last}}^{(t)}$(□□□□□□)□$y_{\mathrm{last}}^{(t)}$(□□□□□□□□□)□

□□,RNN□□□□□□□□□□□□□□□□□,□□□□□(Conditional Random Field,CRF)□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□(named entity extraction)□□,□□□□□□□□□□,□□□□□□□[□“I go to San Francisco”(□□□□□)]□□□□□□□□□□□□□□□:{□□

(location),□□(name),□□□□(company name),□□(other)}□□□□□□□□□□(□□□)□□□□□□,□“□□□□□□□□□□□”□“□□□□□□□□□□□”,□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□,□□□□□□□□□□□□□□□

CRF□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□RNN□□□□□□□,□□□□□□□□□,□□CRF□□□□□□□□□(□□□□□□□□)□□□□□,□□□□□□□□□□□□□□□□□

7.7 □□□□□□

□□□□□□□(sequence-to-sequence learning,seq2seq)□□□□□□□□□□□□□□seq2seq□,X_i□Y_i□□□□□□□□seq2seq□□□□□□□□□□(□□□□□□□□,□□□□□□□□□□□)□□□□□(□□□□□□□,□□□□□)□□□□□□□□□□□□

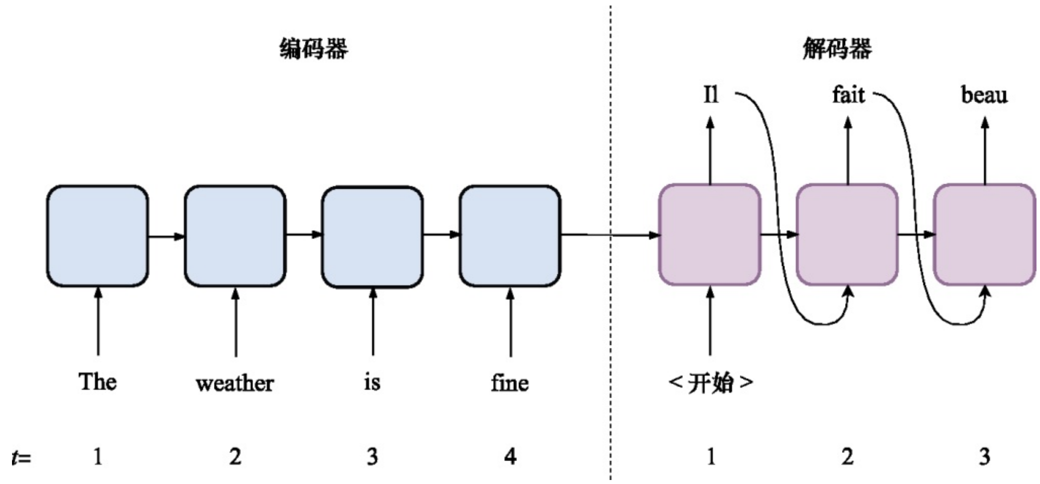
□□□□□□,□□□□□□□□□□seq2seq□□□□□□□□□□seq2seq□□□□□□□□□□:□□□□□(encoder)□□□□□□(decoder)□

[illegible]seq2seq[illegible],[illegible],[illegible]RNN,[illegible]CNN[illegible]([illegible]RNN[illegible])[illegible],[illegible]([illegible])[illegible],[illegible]([illegible])[illegible](embedding)[illegible]

[illegible],[illegible],[illegible],[illegible]$\boldsymbol{x}^{(0)}$[illegible](start of sequence)([illegible]0),[illegible]$\boldsymbol{y}^{(1)}$[illegible],[illegible]$\boldsymbol{x}^{(0)}$[illegible],[illegible]$\boldsymbol{y}^{(1)}$[illegible]$\boldsymbol{x}^{(1)}$[illegible],[illegible]$\boldsymbol{y}^{(t)}$[illegible]$\boldsymbol{x}^{(t)}$[illegible];[illegible],[illegible]6[illegible],RNN[illegible]:[illegible]$\boldsymbol{y}^{(t)}$,[illegible]$\boldsymbol{x}^{(t)}$[illegible]

[illegible]

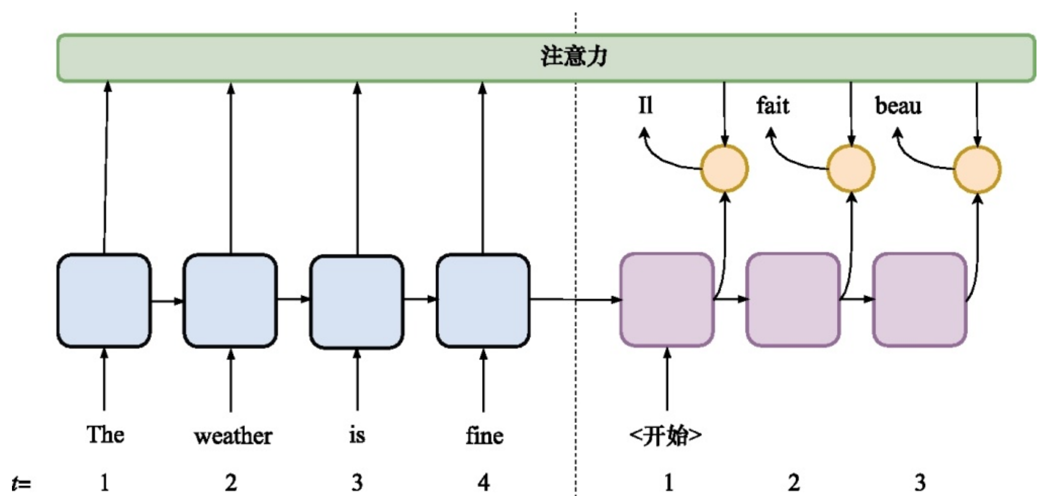
[illegible]seq2seq[illegible]7.4[illegible](attention)[illegible],[illegible]([illegible]RNN[illegible],[illegible])[illegible]RNN[illegible],[illegible]



□7.4　□□□□□seq2seq□□

□:□□,□□□□□□□□□□□□□□□7.4□,□□□□□□□□□□□

□7.5□□□□□□□□seq2seq□□



□7.5　□□□□□□□□seq2seq□□

seq2seq□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□

□□□□□,□□□□□□□□,□□□□□□□□□□□□□□

7.8 主动学习

主动学习(active learning)是另一种有趣的监督学习范式,通常应用于获取标注样本成本很高的情况下。在医学或金融领域中经常出现这种情况,此时可能需要专家的意见对患者或客户的数据进行标注。这个想法是,我们用一个相对较小的有标注样本集开始学习,并使用大量无标注样本,然后只标注那些对模型质量贡献最大的样本。

主动学习有多种策略,这里我们只讨论以下两种策略:

- 基于数据密度和不确定性的策略
- 基于支持向量的策略

前一种策略中,我们使用已有的有标注样本训练当前模型f,对于剩下的所有未标注样本(或者为了节省时间,只对其中部分随机样本)计算每个未标注样本$\boldsymbol{x}$的重要性分数:密度($\boldsymbol{x}$)•不确定性$f(\boldsymbol{x})$。密度反映了$\boldsymbol{x}$周围有多少样本,而不确定性反映了模型f对$\boldsymbol{x}$的预测有多不确定。在使用sigmoid的二元分类中,预测分数越接近0.5,预测就越不确定。在SVM中,样本越接近决策边界,预测就越不确定。

在多元分类问题中,可以用熵作为典型的不确定性度量:

$$H_f(\boldsymbol{x}) = -\sum_{c=1}^{C} \Pr\left(y^{(c)};\quad f(\boldsymbol{x})\right) \ln\left[\Pr\left(y^{(c)};\quad f(\boldsymbol{x})\right)\right]$$

其中,$\Pr(y^{(c)};f(\boldsymbol{x}))$是模型$f$给类别$y^{(c)}$分配给$\boldsymbol{x}$分类的概率。可以看出,如果对于每个$y^{(c)}$,$f(y^{(c)})=\frac{1}{C}$,那么模型最为不确定,熵也达到最大值。另一方面,如果对于某个$y^{(c)}$,$f(y^{(c)})=1$,那么模型确定分类为$y^{(c)}$,熵为0。

样本$\boldsymbol{x}$的密度可以通过求$\boldsymbol{x}$和k个最近邻样本的平均距离来得到(k是一个超参数)。

[illegible],[illegible],[illegible],illegible,[illegible])[illegible]

[illegible],[illegible]SVM[illegible],[illegible],[illegible],[illegible],illegible[illegible]

[illegible],[illegible]“[illegible]”(query by committee)[illegible],[illegible]

7.9 [illegible]

[illegible](semi-supervised learning,SSL)[illegible],[illegible];[illegible],[illegible],[illegible]

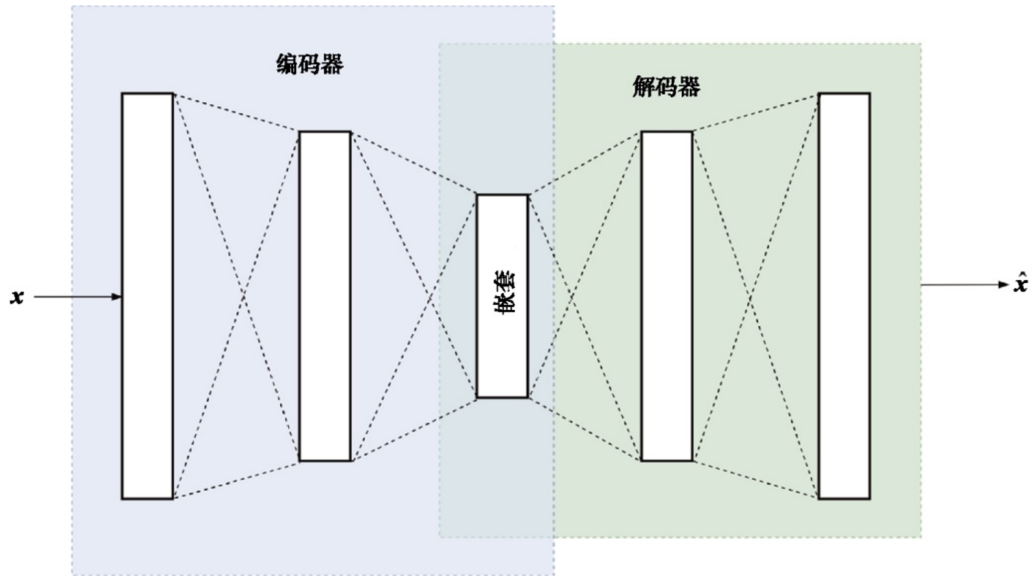
[illegible],[illegible],[illegible],[illegible]SSL [illegible](self-learning)[illegible],[illegible],[illegible]$\boldsymbol{x}$[illegible]([illegible]),[illegible],[illegible],[illegible],[illegible]m[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]MNIST([illegible],[illegible]0[illegible]9),[illegible]10[illegible]([illegible]100[illegible])[illegible]MNIST[illegible]70 000[illegible](60 000[illegible],10 000[illegible])[illegible](ladder network)[illegible],[illegible]

[illegible]-[illegible],[illegible],[illegible]($\boldsymbol{x}$,$\boldsymbol{x}$)[illegible]$f(\boldsymbol{x})$[illegible]$\hat{x}$[illegible]$\boldsymbol{x}$[illegible]

[illegible],[illegible],[illegible](bottleneck layer)[illegible]D[illegible];[illegible]D[illegible],[illegible]10[illegible]MNIST[illegible]7.6[illegible],[illegible]([illegible])[illegible]([illegible],[illegible]S[illegible])[illegible]



□7.6 □□□□

□□□□□□□□□,□□□□:

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{x}_i-f\left(\boldsymbol{x}_i\right)\right\|^2$$

□□,$\|\boldsymbol{x}_i - f(\boldsymbol{x}_i)\|$□□□□□□□□□□□□

□□□□□□□□(denoising autoencoder)□□□□□□□□□$\boldsymbol{x}$□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□0□1□□□□□□,□□□□□□□□□□□□□□□(normal Gaussian noise)□□□□□□$\boldsymbol{x}$□□□□□j,□□□$n^{(j)}$□□□□□□□□:

$$n^{(j)} \sim \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(-\mu)^2}{2\sigma^2}\right)$$

□□,~□□□□□□“□…□□□”,π□□□□3.141 59…,□□μ□□□□□□□□□□□□□□□□$\boldsymbol{x}^{(j)}$,□$\boldsymbol{x}^{(j)}+n^{(j)}$□□□□□

□□(□□softmax□□□□)□□□□□□□□□□□□□□□□□□□l□□□□□□□□□□□□l□,□□C_d^l□□□□□□□□□□□(□□□□□□□□□□)□□□□□□□□□□□□□□,□□□□□□□C_c□□□□□□□□□□□□□□□(□□□□□□□□□□□□)□□□,□□□□□□□□□□□□□□□(minibatch stochastic gradient descent with backpropagation)□□□□□□□, $C_c + \sum_{l=1}^{L} \lambda_l C_d^l$ (□□□□□□□□□□□□)□□□□□□,□□□λ_l□□□□□□□□□□□□□□-□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□□□□□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□□$\boldsymbol{x}$□□□□,□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□,□□□□□□□□(□□□□□9□□□□□□□)□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□S3VM,□□SVM□□□□□□□□□□□□□□□□□□SVM□□,□□□□□□□□□□□□□S3VM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

7.10 □□□□□

□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□(one-shot learning),□□□□□□□□□□□□□□□□□□□□□□,□□□□
□□□□□□,□□,□□□□□□□□□□
□□□□□□□□□□□□□

□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□(□□□□□□□)
□□(□□□□□□□)□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,
□□□□□□□□□□□□□□□

□□□□□□□□□□□□,□□□□□□□□□□□□(Siamese Neural Network,SNN)□
□□SNN□□□□□□□□□□□□□□□□□□□,□□□CNN□RNN□MLP□□□□□□□□□□□□□□□□□;□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□“□□□□”□“□□□
□”,□□□□□□□□□□□□□□□□□□

□□□□□□SNN,□□□□□□□□□□□□(triplet loss)□□□□□□□□□,□□□□□□3□□□□□
□□□□□□□□:□□*A*(□□)□□□*P*(□□)□□□*N*(□□)□□□,*A*□*P*□□□□□□□□□□□□;*N*□□□□
□□□□□□□□,□□□□□□□□□□□□□□(A_i,P_i,N_i)□

□□□□□□□□□□□□□□□*f*□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□*i*□□□□□□□□
□□:

$$\max\left(\|f(A_i)-f(P_i)\|^2-\|f(A_i)-f(N_i)\|^2+\alpha,0\right)$$

(7.3)

这个损失函数的总体形式可以写成:

$$\frac{1}{N}\sum_{i=1}^{N}\max\left(\|f\left(A_i\right)-f\left(P_i\right)\|^2-\|f\left(A_i\right)-f\left(N_i\right)\|^2+\alpha,0\right)$$

其中,α是一个非负数学习参数。由于A和P是同一个人的照片,$\|f(A)-f(P)\|^2$应该非常小;对于不是同一个人的两张照片的编码,$\|f\left(A_i\right)-f\left(N_i\right)\|^2$应该非常大。公式的意义在于,$m=\|f\left(A_i\right)-f\left(P_i\right)\|^2-\|f\left(A_i\right)-f\left(N_i\right)\|^2$应该为负值才对,但是设置一个合理的间隔距离,即加上α后,如果使得m还是负值的,就认为损失值为零。为了使得损失值为零,需要m为负值,且大于α的绝对值,这样就起到了更好的区分作用。

在这个过程中,如何选择N,是一个重要的问题,最好是那种难以识别的人作为反例,即A与P的距离应该和N的距离接近。如果N是随机选择的图片,训练后的模型的区分能力就会弱很多。但是由于都是相近的照片,训练难度就会加大,训练时间更长。

相对于传统的SNN,三元组损失函数的训练方式更为有效,也就是采用端到端的网络,先训练一个CNN网络或者主干网络,提取图像特征后,输入到损失函数中进行三元组A、P、N,进行公式7.3的训练。最终输出的特征值就是人脸特征模型,也就是编码;整个过程就是人脸特征模型的训练过程。

对于训练好的人脸特征模型,如果有一张新的照片在模型中进行人脸识别,进行前向传播后,就会得到对应的编码,这种方式被称为“一对多识别”。如果我们有两个以上的人脸特征编码,就可以对比编码之间的距离,来判断是不是同一个人,如果在训练或者识别过程中,遇到照片的编码差异小于一定阈值,就会认为两张照片A和$\hat{A}$属于同一个人。因此,只需要比较$\left\|f\left(A\right)-f\left(\hat{A}\right)\right\|^2$是否小于一个阈值$\tau$。

7.11 [illegible]

[illegible](Zero-Shot Learning,ZSL)[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]?

[illegible],[illegible]**x**,[illegible]**y**[illegible],[illegible]:[illegible]y_i[illegible]y_k[illegible],[illegible],y_i[illegible]Paris([illegible])[illegible]y_k[illegible]Rome([illegible]),[illegible];[illegible],[illegible]y_k[illegible]potato([illegible]),y_i[illegible]y_k[illegible](word embedding),[illegible][1][illegible]

[illegible],[illegible],[illegible]4[illegible]([illegible],[illegible]50[illegible]300[illegible]),[illegible]4[illegible]:[illegible](animalness)[illegible](abstractness)[illegible](sourness)[illegible](yellowness)[illegible],[illegible]bee([illegible])[illegible][1,0,0,1],[illegible]**yellow**([illegible])[illegible][0,1,0,1],[illegible]**unicorn**([illegible])[illegible][1,1,0,0][illegible]

□□,□□□□□□,□□□□□□□□□□□□□i□□□y_i□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{x}$□□□,□□□□□□f,□□□□□□$\hat{y}$□□□,□□□□□□□□□□□□□$\hat{y}$□□□□□□□□□□□

□□□□□□□?□□□□□(zebra)□□,□□□□□□□□□□□□□□□□□□□□□□□(clownfish),□□□□□□□□□□□□□□□□□□□□3□□□□□□□□□□,CNN□□□□□□□□□□□□□□□□□□□□□□□□□□□□(tiger),□□□□□□□□□□□□□□□□,CNN□□□□□□□□□□□□□□□□(mammalness)□□□□(orangeness)□□□□□□□□(stripeness)□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□

---

[1] □□10□□,□□□□□□□□□□□□□□□□

[illegible]8[illegible] [illegible]

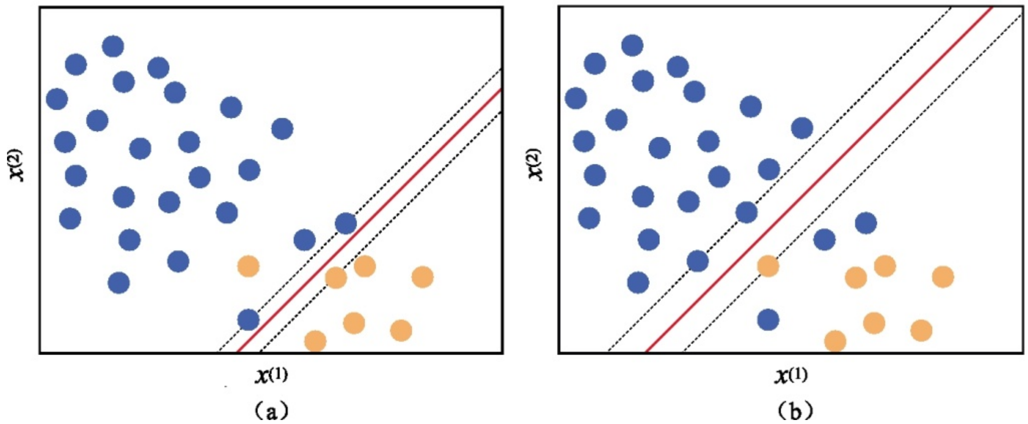
[illegible]“[illegible]”[illegible],[illegible],[illegible]

8.1 [illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible](soft margin)[illegible]SVM,[illegible],[illegible],[illegible]

SVM[illegible],[illegible]“[illegible]”[illegible],[illegible]8.1(a)[illegible]

[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible]8.1(b)[illegible]



□8.1　□□□□□□□□□□

(a)□□□□□□□□□□;(b)□□□□□□□□□□□□

□□SVM□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□□□□□□□□□□(oversampling)□□□□□□,
□□□□□□□□□□,□□□□□□□□□

□□□□□□□□□□□□(undersampling),□□□□□□□□□□□□□□□□□□□□
□□

□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□,□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□(synthetic minority
oversampling technique,SMOTE)□□□□□□□□□(adaptive
synthetic sampling method,ADASYN)□

SMOTE□ADASYN□□□□□□□□□□□□□□□□□□□□$\boldsymbol{x}_i$,□□□□□□□□k□□
□(□S_k□□),□□$\boldsymbol{x}_i+\lambda(\boldsymbol{x}_{zi}-\boldsymbol{x}_i)$□□□□□□□□$\boldsymbol{x}_{\text{new}}$□□□,$\boldsymbol{x}_{zi}$□□$S_k$□□□□□□□□□□□
□□□□□□□□□□□□λ□□□0□1□□□□□□□

SMOTE[illegible]ADASYN[illegible]$\boldsymbol{x}_i$[illegible]ADASYN[illegible],[illegible]$\boldsymbol{x}_i$[illegible]S_k[illegible],[illegible],[illegible]

[illegible]

8.2 [illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]3[illegible]:[illegible](averaging),[illegible](majority vote)[illegible](stacking)[illegible]

[illegible],[illegible],[illegible](base model),[illegible]$\boldsymbol{x}$,[illegible],[illegible]

[illegible]$\boldsymbol{x}$[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible]f_1[illegible]f_2,[illegible]$\hat{\boldsymbol{x}}_i=[f_1(\boldsymbol{x}),f_2(\boldsymbol{x})]$[illegible]$\hat{y}_i=y_i$,[illegible]$(\hat{\boldsymbol{x}}_i,y_i)$[illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]

[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible]“[illegible]”[illegible],[illegible],[illegible]SVM[illegible],[illegible]SVM,[illegible]

8.3 [illegible]

[illegible],[illegible],[illegible],[illegible];[illegible],[illegible](standardize),[illegible](normalize)[illegible][0,1][illegible]

[illegible]([illegible],[illegible])[illegible]CNN[illegible]RNN,[illegible](one-hot encoding)[illegible],[illegible](word embedding)[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible]seq2seq[illegible],[illegible],[illegible](Google Scholar)[illegible](Microsoft Academic)[1][illegible],[illegible]GitHub([illegible])[illegible],[illegible]

[illegible],[illegible],[illegible];[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]([illegible])[illegible],[illegible],[illegible],[illegible]([illegible]),[illegible],[illegible],[illegible]

8.4 [illegible]

[illegible],[illegible]L1[illegible]L2[illegible],[illegible]:[illegible](dropout)[illegible](early stopping)[illegible](batch normalization)[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]0[illegible]1[illegible],[illegible]

[illegible],[illegible],[illegible]4[illegible],[illegible],[illegible],[illegible]*e*[illegible],[illegible]:[illegible],[illegible]

对训练集数据进行多轮学习时,第一轮的学习进度可能看起来很快并且十分准确,但是经过多轮学习后就可能会出现过拟合,此时模型的性能会开始下降。为了避免这种情况,可以设置在训练过程中每当模型在验证集上有所进步时,就会生成一个模型的检查点(checkpoint)以记录当前的模型参数以及训练状态;当训练结束时选择最优的检查点作为模型。

标签平滑也是一种常用的策略。标签平滑就是在训练时对数据的标签引入噪声,从而降低模型对于标签的置信度,防止模型过度拟合训练数据。例如,对于某个二分类问题,原来的标签是0或者1,经过平滑后可能会变成在两者之间的数值。

另外,在训练数据不足以训练模型的时候,需要使用一些方法扩充训练数据,如数据增强(data augmentation)方法。对于文本数据而言,可以进行同义词替换,或者随机删除句子中的词语,或者随机交换两个词语的位置,或者随机插入同义词。对于图像数据而言,可以进行旋转、翻转、裁剪、平移、缩放、添加噪声等操作,使得模型在训练时获得更多的不同样本数据,从而提升模型的泛化能力。

8.5 多模态学习

在实际中,有很多数据是多模态(multimodal)的。例如,一段视频中包含图像、声音和文字等信息,因此需要将多种模态的信息结合起来。

在多模态学习中,需要首先将各种模态的数据进行表示,然后再对各种模态的数据表示进行融合以进行学习,这就涉及数据的特征提取与融合,以及如何利用这些融合后的特征进行学习。

有一种简单的方法是将各个模态的特征串接起来,然后输入模型中对其进行学习的方法(即早期融合或特征级融合),例如,对于同时包括图像和文本的样本数据,如果图像的特征向量为[$i^{(1)}$,$i^{(2)}$,$i^{(3)}$]、文本的特征为[$t^{(1)}$,$t^{(2)}$,$t^{(3)}$,$t^{(4)}$],则串接后的特征为[$i^{(1)}$,$i^{(2)}$,$i^{(3)}$,$t^{(1)}$,$t^{(2)}$,$t^{(3)}$,$t^{(4)}$]。

另一种方法是先分别对各种模态的数据进行学习得到其各自的高层表示,再对这些高层表示进行融合学习,例如用CNN获取图像的特征表示,然后用RNN获取文本的特征表示,再将两者进行融合以完成后续的任务。

率;CNN可以局部地构建,而RNN可以提供构建的平滑模型,对于计算某个词元的邻接词元,可以进行调整;此外我们可以很自然地,将softmax和sigmoid等处理技术融入这些网络模型中,这样就可以处理某个特定的专业任务。

8.6 联合学习法

通常由一个非常大1的训练语料库训练的词元向量可以用于很多专业领域。然而对于特定的专业任务来说,如具体的词元分类(如标签,tag),一些重新训练中还需要不断调整词元向量。

例如,对于需要进一步精确化的任务,还可以采用下述方法来完成一项工作:对全部词元进行分类,如果没有对于特定任务的语料库进行训练,那么对于相近意义的词元进行分类时,例如"很好""好"以及"不错"这样的词元分在不同的类别中进行研究,因为它们常常出现在不同领域的上下文之中。

在目前的工作中,可以用下述方法进行联合学习词元,如下1个方法用于训练以查询的结果进行排序的联合学习过程,首先对词元形成一个自然语言处理的单元模型中,也就是我们所构成的词元向量组,然后在经过一个非线性变换,如ReLU,把它们经过一个神经网络后,获得一个用于查询的搜索结果的排序函数C_1。然后对于其他的查询结果再进行排序,这样就可以由另外一个神经网络的处理形成之后再使用softmax对输出处理,可以得到另外一个排序函数C_2(即交叉熵函数)。

现在我们所讨论的联合学习的目的是要将这两个函数组合,使得它们能同时对两种需要的训练进行优化,从而使得我们构造的模型,能够同时完成这两项任务,这样我们只需要选取一个在(0,1)区间中的γ,这样联合学习的目标就是$\gamma C_i + (1-\gamma) C_2$求出最小值时的参数,然后再根据实际情况调整参数$\gamma$的值。

8.7 本章小结

[illegible],[illegible](transfer learning)[illegible][illegible],[illegible],[illegible][illegible],[illegible],[illegible][illegible],[illegible]

[illegible]([illegible])[illegible],[illegible],[illegible][illegible]:[illegible],[illegible]:[illegible][illegible]

[illegible],[illegible]:

(1)[illegible]([illegible])[illegible]

(2)[illegible]([illegible])[illegible]

(3)[illegible],[illegible]

(4)[illegible],[illegible]

(5)[illegible]“[illegible]”[illegible]

(6)[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]([illegible]),[illegible]

[illegible],[illegible],[illegible][illegible],[illegible],[illegible]([illegible])[illegible],[illegible]1 000

[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible],[illegible]

8.8 [illegible]

[illegible],
[illegible],[illegible]

[illegible](analysis of algorithm)[illegible],[illegible]**O**[illegible](Big O notation)[illegible],[illegible],[illegible]

[illegible],[illegible]1[illegible]N[illegible]S[illegible](Python[illegible],[illegible]):

```python
def find_max_distance(S):
    result = None
    max_distance = 0
    for x1 in S:
        for x2 in S:
            if abs(x1 - x2) >= max_distance:
                max_distance = abs(x1 - x2)
                result = (x1, x2)
    return result
```

[illegible],[illegible]S[illegible],[illegible]S[illegible],[illegible]N^2[illegible]comparison([illegible])[illegible]abs([illegible])[illegible]assignment([illegible])[illegible],[illegible]([illegible])[illegible]

$5N^2$(□□□□□□1□comparison□2□abs□2□assignment□□□□□□□□□□□□□□□□□□□□□□□,□□□□O□□□□□□□□□□□□□□$O(N^2)$□□□,□□□□□5,□□□□

□□□□□,□□□□□□□□□□□□:

```
def find_max_distance(S):
    result = None
    min_x = float("inf")
    max_x = float("-inf")
    for x in S:
        if x < min_x:
            min_x = x
          if x > max_x:
            max_x = x
    result = (max_x, min_x)
    return result
```

□□□□□□□□,□□□□□□□□□S□□□□□□,□□□□□□□□□$O(N)$□□□,□□□□□□□□□□□□□□□□□(more efficient)□

□□□□□□□□□□□□□□□□□□(polynomial)□,□□□□□□□□□□,$O(N^2)$□$O(N)$□□□□□□,□□N□N^2□□□N□1□□□□□2□□□□□□□,□□□□□□□□,□□$O(N^2)$□□□□□□□□□□□□□□□,□□□□□□□□□$O(\log N)$□□□□

□□□□□□□,□□□□□□,□□□□□□□□□□□□(avoid using loops whenever possible),□□□□□□□□□□□□□□□□□□Python□□□,□□□wx,□□□:

```
import numpy
wx = numpy.dot(w,x)
```

□□□:

```
wx = 0
for i in range(N):
    wx += w[i]*x[i]
```

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□(set)□□□□□(list)□□□□□□Python□,□□□□□□□□□□□*x*□□□*S*□□□,□□□*S*□□□□□□□□□,□□*S*□□□□□□□□□□

□□□□□□Python□□□□□□□□□□□□□□(dict)□□□□□□□□□□□□□□□□(hashmap)□□□□□□□□□□□□□□(key-value pair)□□□,□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Python□□numpy□scipy□□scikit-learn,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□(method)□C□□□□□

□□□□□□□□□□□□□□□□□□□□□□,□□□□□(generator)□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□Python□□cProfile□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□:

- □□multiprocessing□□□□□□□□
- □□PyPy□Numba□□□□□□Python□□□□□□□□□□□□□□□□□□

---

[1] □□□:□□□□□□□□□□□□□□□□□

□9□ □□□□□

□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

9.1 □□□□

□□□□(density estimation)□□□□□□□:□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□(probability density function,pdf)□□□□□□□□□□□□□□□,□□□□□□□□□□□□□7□□,□□□□□□□□pdf□□□□□□□□□□□□□,□□□□□□□□□(parametric)□□,□□□□□□□□□□□□□□(multivariate normal distribution,MVN)□□□□□□□□□□□□□□□□□□□□□□□□□□□MVN□□,□□□□□□□□□□□□□□□□,□□□□□□□,□□□□□,□□□□□□□□(nonparametric)□□□□,□□□□□□□□□□□□□□□

□$\{x_i\}_{i=1}^{N}$□□□□□□□□(□□□□□□□),□□□□□□□pdf□□f,□□□i=1,…,N□□□$x_i \in \mathbb{R}$□□□,□□□□□f□□□□□□f□□□□$\hat{f}_b$□□□□:

$$\hat{f}_b\left(x\right) = \frac{1}{Nb}\sum_{i=1}^{N} k\left(\frac{x - x_i}{b}\right) \tag{9.1}$$

□□,*b*□□□□□□□□□□□,*k*□□□□□7□□□,□□□□□□:

$$k(z)=\frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)$$

□□□□*b*□,□□*f*□□□□□$\hat{f}_b$□□□□□□□□□□□□□(mean integrated squared error,MISE)□□:

$$\mathrm{MISE}\,(b)=\mathbb{E}\left[\int_{\mathbb{R}}\left(\hat{f}_b\,(x)-f\,(x)\right)^2\mathrm{d}x\right]$$

(9.2)

□□9.2□,□□□□□pdf *f*□□□$\hat{f}_b$□□□□□□□$\int_{\mathbb{R}}$□□□□□□□□□$\sum_{i=1}^{N}$□□□,□□□□□□$\mathbb{E}$□□□□□□$\frac{1}{N}$□

□□□□□□□,□$(\hat{f}_b\,(x)-f\,(x))^2$,□□□□□□□□□□□□□$\mathbb{E}$□□□□□*b*□□□□□$\{x_i\}_{i=1}^{N}$□□□□□□□□,□□$\hat{f}_b$□□□□□□□□□□□,□□□pdf *f*□□□□□(ℝ□□)□□□

□□□9.2□□□□□:

$$\mathbb{E}\left[\int_{\mathbb{R}}\hat{f}_b^2\,(x)\,\mathrm{d}x\right]-2\mathbb{E}\left[\int_{\mathbb{R}}\hat{f}_b\,(x)\,f\,(x)\,\mathrm{d}x\right]+\mathbb{E}\left[\int_{\mathbb{R}}f(x)^2\mathrm{d}x\right]$$

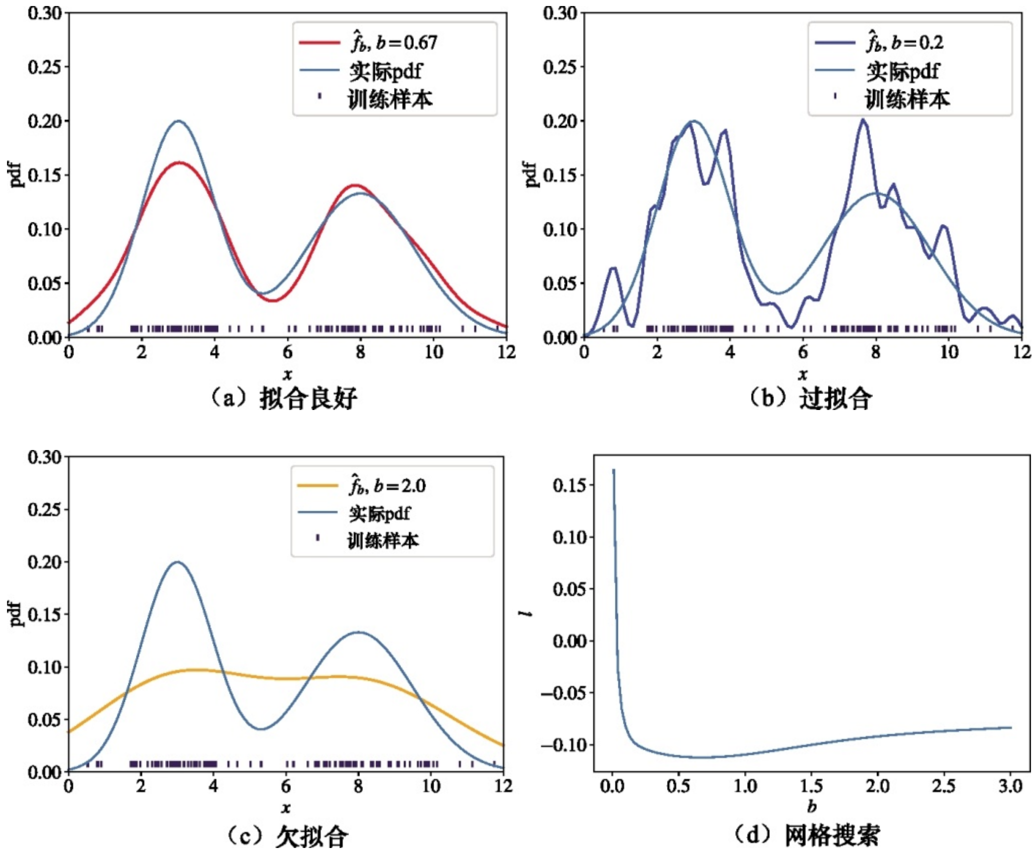
□□□□□□□□□*b*,□□□□□□□□□□□$\int_{\mathbb{R}}\hat{f}_b^2\,(x)\,\mathrm{d}x$;□□□□□□□□□(cross-validation)□□$-\frac{2}{N}\sum_{i=1}^{N}\hat{f}_b^{(1)}\,(x_i)$,□□$\hat{f}_b^{(1)}$□*f*□□□,□□□$x_i$□□□□□□□

$\sum_{i=1}^{N} \hat{f}_b^{(1)}(x_i)$□□□□□□□□□□□□□(leave one out estimate)□□□□□□□□□□□□,□□□□(fold)□□□□□□□□□□□□□□□□,$\int_{\mathbb{R}} \hat{f}_b(x) f(x)\,\mathrm{d}x$□(□□□□□$a$)□□□$\hat{f}_b$□□□□,□□$f$□□□pdf□□□□□□□□□□□□□$\mathbb{E}[a]$□□□□□

□□□,□b□□□□b^*□□□□□□□□□□□□:

$$\int_{\mathbb{R}} \hat{f}_b^2(x)\,\mathrm{d}x - \frac{2}{N} \sum_{i=1}^{N} \hat{f}_b^{(1)}(x_i)$$

□□,□□□□□□□□□□□□□□□□D□□□□□$\boldsymbol{x}$,□9.1□□□□□x-x_i□□□□□□□□||$\boldsymbol{x}$-$\boldsymbol{x}_i$||□□□9.1□,□□□□100□□□□3□□□□b□□□□□pdf□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□b^*□



（a）拟合良好 （b）过拟合

（c）欠拟合 （d）网格搜索

□9.1　□□□□□

9.2 □□

□□(clustering)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□,□□□□□□□□□□□□,□□□□□□□□□□□

□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□
□□□□□□□□□□□□□,□□□□□□□□□□□□

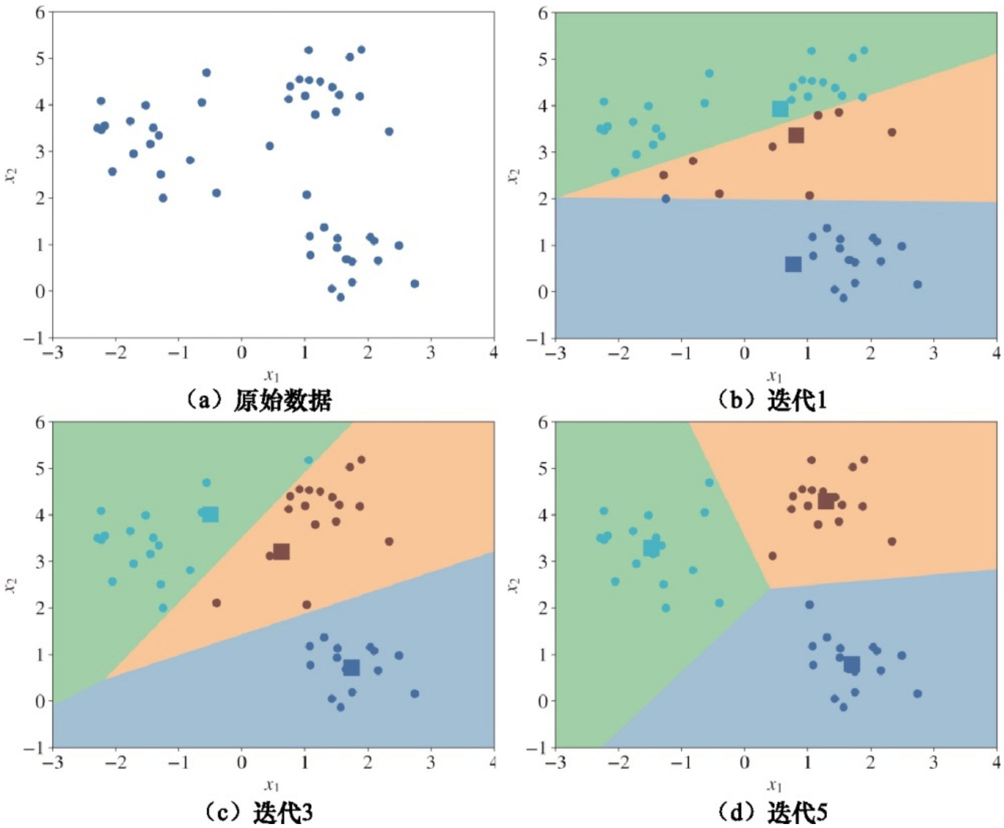
9.2.1 k□□

k□□(k-mean)□□□□□□□□□□,□□□*k*□□□□(cluster)□□□□□,□
□□□□□□□*k*□□□□,□□□(centroid)□□□,□□□□□,□□□□,□□□□□***x***□
□□□c□□□□□□□□□,□□□□□□□□□□□□□(□□□□□□□□□□)□□□,□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□□□,□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□k□□□□□□□□□□□□k□□□□□□,□□
□□□□□□□□□□□□□□□□

□□□k□□□□□□9.2□□□□9.2□,□□□□□□□□□;□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□



□9.2 □k=3□,k□□□□□□□

□□□k□□□□□□□□□,□□□□□□□□□□□□□□□□k□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□”,□□□□□□□□□□□□□□□□,□□□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□

9.2.2 DBSCAN□HDBSCAN

DBSCAN□□□□□□□□□□,□□□k□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□:ϵ□n□□□□□,□□□□□□□□□□□□□□$\boldsymbol{x}$,□□□□□□□□1□□□,□□□□□□□□□$\boldsymbol{x}$□□□□□□□□□□ϵ□□□□□□□□□□n,□□□□ϵ□□□□□□1□□□□□□□□□1□□□□□□,□□□□□ϵ□□□□□□□□□n□□□□ϵ□□,□□ϵ□□□□□□□□1□□□□□□

[illegible]1[illegible],[illegible],[illegible]2[illegible],[illegible],[illegible]ϵ[illegible]n[illegible],[illegible]

DBSCAN[illegible],[illegible],k[illegible]DBSCAN[illegible],[illegible]([illegible]ϵ)[illegible],[illegible]ϵ[illegible],[illegible]

HDBSCAN[illegible]DBSCAN[illegible],[illegible]ϵ[illegible],[illegible]

HDBSCAN[illegible]n,[illegible],[illegible]HDBSCAN[illegible]:[illegible],[illegible]k[illegible]HDBSCAN[illegible],[illegible]HDBSCAN[illegible]

9.2.3 [illegible]

[illegible],[illegible],[illegible],[illegible],[illegible][1][illegible]

[illegible](prediction strength)[illegible],[illegible],[illegible],[illegible]N_{tr}[illegible]S_{tr}[illegible]N_{te}[illegible]S_{te}[illegible]k[illegible],[illegible]S_{tr}[illegible]S_{te}[illegible]C[illegible]$C(S_{tr},k)$[illegible]$C(S_{te},k)$[illegible]

□A□□□□□□□□□□□$C(S_{tr},k)$□A□□□□□□□□□□,□□□□□□□□□□□□,□□□□

□□□□□□k□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□k□□□□□□,□□9.2

□□□

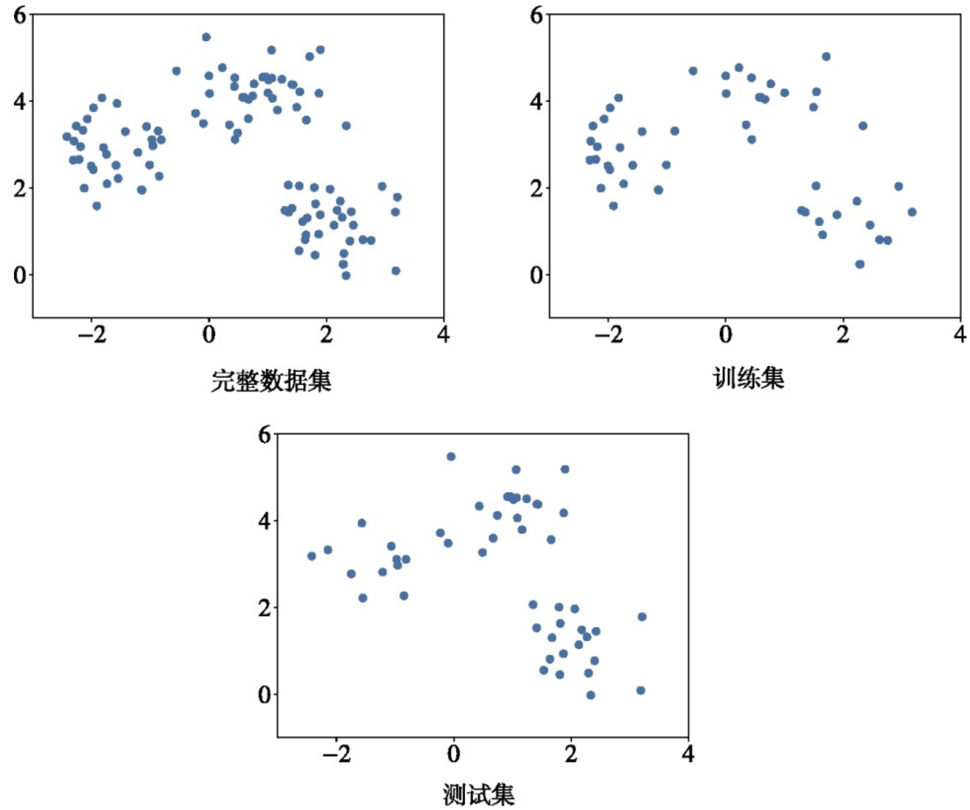
□□,□□□□□□□$N_{te}\times N_{te}$□□□□□□□(co-membership matrix)$\boldsymbol{D}[A,S_{te}]$:□□□□□□□□□□□□□$\boldsymbol{x}_i$□$\boldsymbol{x}_{i'}$□A□□□□□□□□□□,$\boldsymbol{D}[A,S_{te}]^{(i,i')}=1$;□□$\boldsymbol{D}[A,S_{te}]^{(i,i')}=0$□

□□,□□□□□□□□□□□□□□□□□□□□□□□□□□k□□□□□□□□A□□□,□□□□□

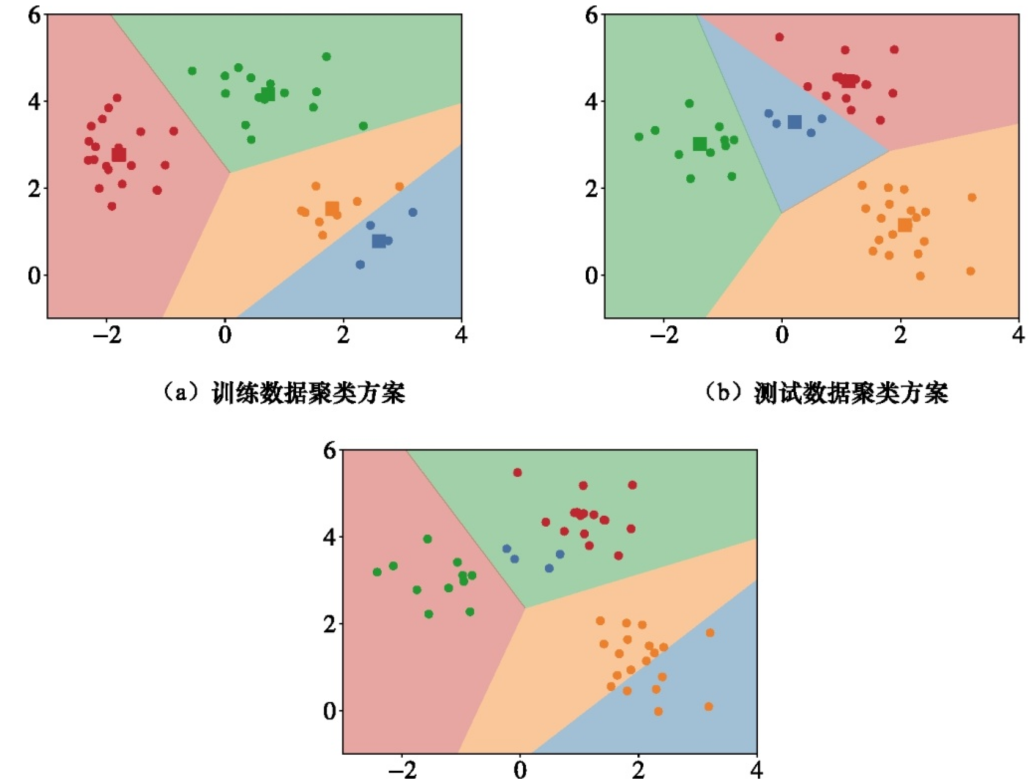
□□□□□□,□□□□□□□□□□□□□□□□□□□A□

□□□□□□□□□□,□□k□□□□□□□□□□□,□□□□□$C(S_{te},k)$□□□□□□□□□□□□,□□□$C(S_{tr},k)$□□□□□□□□□□□□□□□□,□□k□□□□□□(□□□□□),□□□□□□□□□□□□□□□□□□□□□

□9.3□□9.4□□□□□□□□□□□□□□□9.4(a)□□9.4(b)□□□□□□$C(S_{tr},k)$□□□$C(S_{te},k)$□□□□□□□9.4(c)□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□$\boldsymbol{D}[A,S_{te}]$□□□□□0□,□□k=4□□□□□□□



□9.3　□9.4□□□□□□□□□□



（a）训练数据聚类方案

（b）测试数据聚类方案

（c）测试数据重叠于训练数据的聚类方案

□9.4　□k=4□□□□□

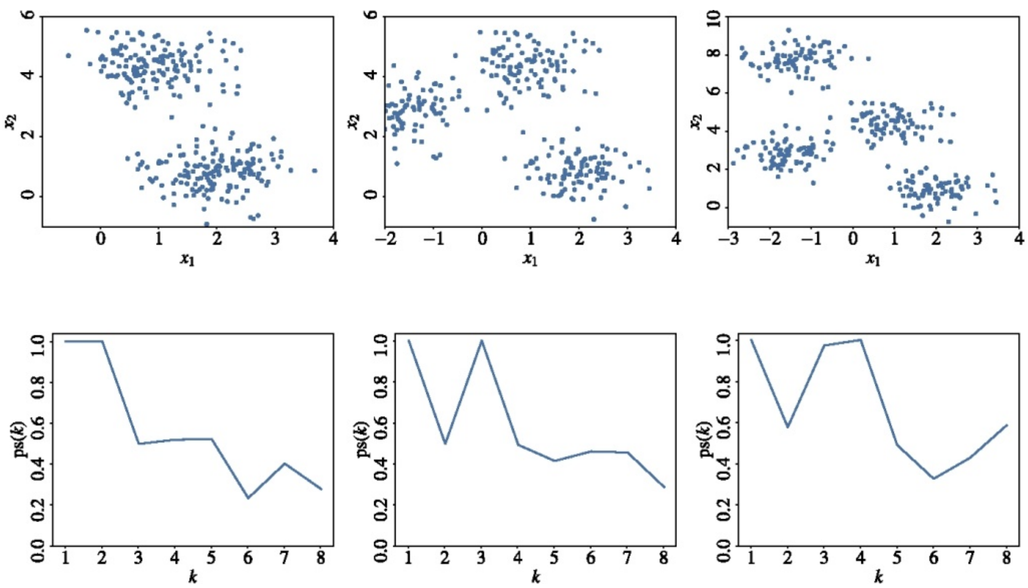
□□□□,k□□□□□□□□□□□□□:

$$\text{ps}(k) \stackrel{\text{def}}{=} \min_{j=1,\ldots,k} \frac{1}{|A_j|(|A_j|-1)} \sum_{i,i' \in A_j} \boldsymbol{D}[A, S_{te}]^{(i,i')}$$

□□,$A \stackrel{\text{def}}{=} C(S_{tr},k)$□□□□□$C(S_{tr},k)$□□$j$□□□,$|A_j|$□□□$A_j$□□□□□

□□□□□□□□$C(S_{tr},k)$,□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□k□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□□□ps(k)□□0.8□□□k□□□k□□□2□□3□□4□□□□□□□□□□□□9.5□□□



□9.5 □□k□□□2□□3□□4□□□□□□□□□□

□□□□□□□□,□k□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□k□,□□□□□□□□□□□□□□□$\overline{\mathrm{ps}}(k)$□

□□□□□□□□□□□□□□□□□□□□(gap statistic)□□□□□□□□□□□□□(elbow method)□□□□□□(average silhouette method)□□□□□□□□□□□□

9.2.4 □□□□□□

DBSCAN□k□□□□□□□□(hard clustering),□□□□□□□□□□□□□□□□□□□(Gaussian mixture model,GMM)□□□□□□□□□□□□□,□□□□□□□□□□(membership score)(HDBSCAN□□□)□□□GMM□□□□□□□□□□□□□□□□□□□GMM□,□□□□□□□□□□□(MND),□□□□□□□MND□□□□□:

$$f_X = \sum_{j=1}^{k} \phi_j f_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}$$

□□,$f_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}$□□□MND j,□ϕ_j□j□□□□□□□□□□□□□j=1,…,k,□□$\boldsymbol{\mu}_j$,$\boldsymbol{\Sigma}_j$□ϕ_j□□□□□□□□□□□(expectation maximization algorithm,EM)□□□□□□□(maximum likelihood)□□□

□□□□□□,□□□□□□□□□□□□□□□□□:k=2□□□,□□□□□□□□□:

$$f(x|\mu, \sigma_1^2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp - \frac{(x-\mu_1)^2}{2\sigma_1^2},$$
$$f(x|\mu, \sigma_2^2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp - \frac{(x-\mu_2)^2}{2\sigma_2^2}$$

(9.3)

□□,$f(x|\mu_1,\sigma_1^2)$□$f(x|\mu_2,\sigma_2^2)$□□□pdf,□□$X=x$□□□□□

□□□□EM□□□□μ_1,σ_1^2,μ_2,σ_2^2,$\phi_1\phi_2$□□□□□ϕ_1□ϕ_2□□□□□□□□□□□□□□□□□□,□□□□□□□□

EM□□□□□□□□□□,□□□□μ_1□σ_1^2□μ_2□σ_2^2□□□□,□□$\phi_1=\phi_2=\frac{1}{2}$[□□□□□,□□□□□$\phi_j(j=1,\ldots,k)$□□$\frac{1}{k}$]□

□EM□□□□□□,□□□□4□□□:

(1)□□□$i=1,\ldots,N$,□□□9.3□□□□x_i□□□□

$$f(x_i|\mu_1,\sigma_1^2) \leftarrow \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp -\frac{(x_i-\mu_1)^2}{2\sigma_1^2},$$
$$f(x_i|\mu_2,\sigma_2^2) \leftarrow \frac{1}{\sqrt{2\pi\sigma_2^2}}\exp -\frac{(x_i-\mu_2)^2}{2\sigma_2^2}$$

(2)□□□□□□□(Bayes’ Rule),□□□□□x_i,□□□□□□□□$j\in\{1,2\}$□□□□$b_i^{(j)}$(□□□□,□□□□□□□□□j□□□□):

$$b_i^{(j)} \leftarrow \frac{f(x_i|\mu_j,\sigma_j^2)\phi_j}{f(x_i|\mu_1,\sigma_1^2)\phi_1+f(x_i|\mu_2,\sigma_2^2)\phi_2}$$

□□ϕ_j□□□□,□□□□□□μ_j□σ_j^2□□□□j□□□□□□□□□□□□□□□□□□□□□□□□□$\phi_1=\phi_2=\frac{1}{2}$□□□:□□□□□□□□□□□□,□□□□□□□□□□□□□□□

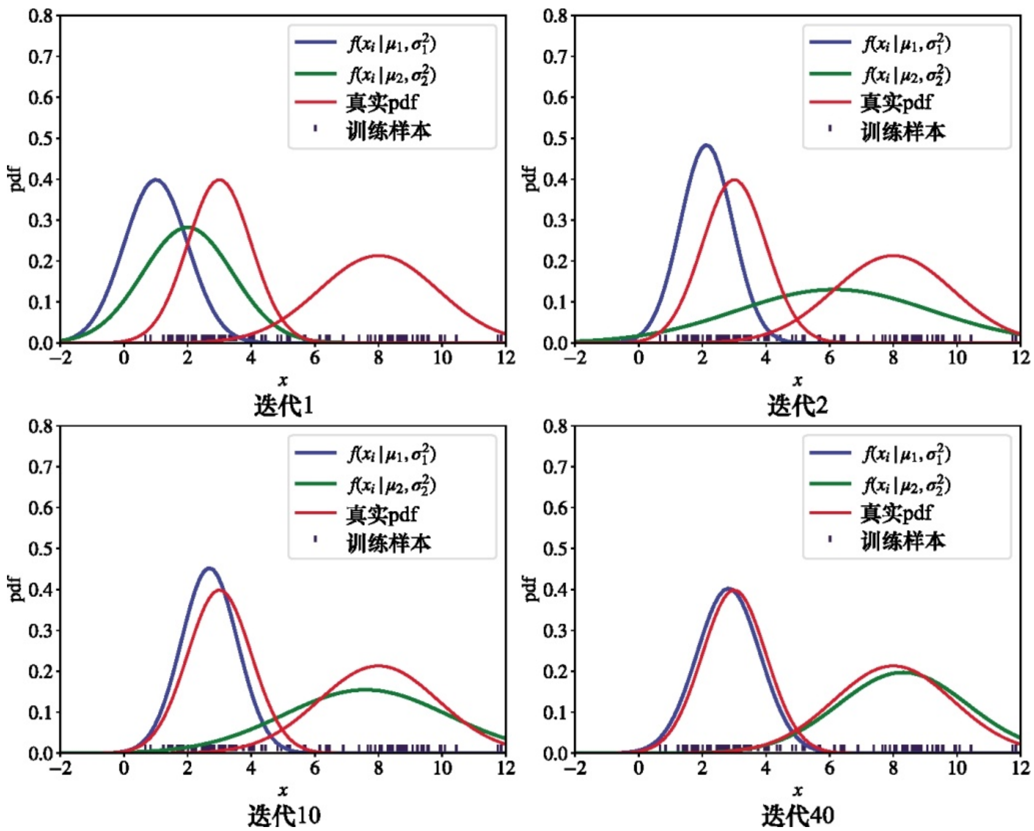
(3)□□□□μ_j,σ_j^2,$j\in\{1,2\}$:

$$\mu_j \leftarrow \frac{\sum_{i=1}^{N} b_i^{(j)} x_i}{\sum_{i=1}^{N} b_i^{(j)}} \quad □ \quad \sigma_j^2 \leftarrow \frac{\sum_{i=1}^{N} b_i^{(j)}(x_i-\mu_j)^2}{\sum_{i=1}^{N} b_i^{(j)}} \qquad (9.4)$$

(4)□□$\phi_j, j \in \{1,2\}$

$$\phi_j \leftarrow \frac{1}{N}\sum_{i=1}^{N} b_i^{(j)}$$

□□□□□□□(1)□(4),□□μ_j□σ_j^2□□□□□□□□:□□□□□□□□□□ϵ□□□□□□□9.6□□□



□9.6　□□EM□□□□□□□(k=2)□□□□□□□□□□□

□□□□□□□EM□□□k□□□□□□□□:□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□GMM□□□□□□x_i□□□j□□□□(soft clustering):x_i□$b_i^{(j)}$□□□□□□□j□□□□□□,□□□□□□9.4□□μ_j□σ_j^2□□□□□□□□□□(□k□□),□□□□□□$b_i^{(j)}$□□□□□□

□□□□□□j□□□μ_j□σ_j^2□□,□□x□□□j□□□□□□□□$f(x_i|\mu_j,\sigma_j^2)$□□□□□

□□□□□□□□□□D□□□(D>1)□□□□□□□σ^2□□□□□□□Σ□□□□□□□□□□□

□□□□□□□(MND)□

k□□□□□□□□□□□□□,□GMM□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

□GMM□,□□□□□□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□□□□

□,□□□□□□□□□□□k□□□□□□□f_{tr}^k□□□,□□□□□□□□□□□□□□□□□k:

$$\arg\max_k \prod_{i=1}^{|N_{te}|} f_{tr}^k(\boldsymbol{x}_i)$$

□□,$|N_{te}|$□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□(spectral clustering)□□□□□

(hierarchical clustering)□□□□□□□□□□□□□□□□□□,□□□□□□□□□,k□

□□HDBSCAN□□□□□□□□□□□□□□□□

9.3 □□□□

[illegible],[illegible],[illegible](GPU),[illegible]:[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]——[illegible]

[illegible]3[illegible],[illegible](Principle Component Analysis,PCA)[illegible](Uniform Manifold Approximation and Projection,UMAP)[illegible](autoencoder)[illegible]

[illegible]7[illegible],[illegible],[illegible]

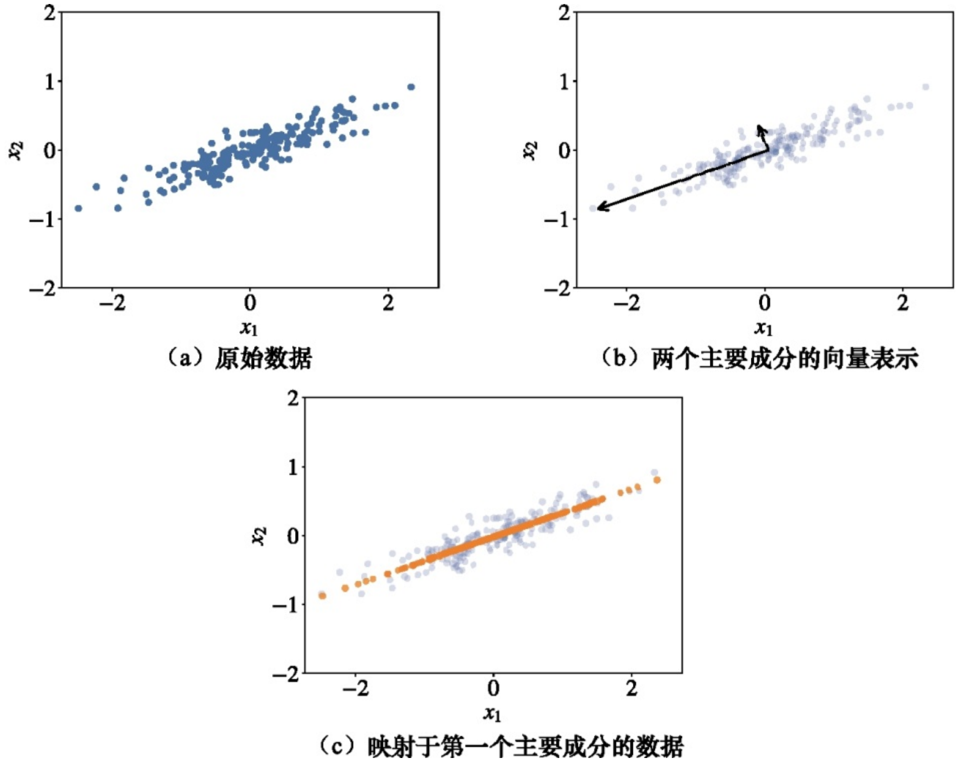
9.3.1 [illegible]

[illegible](PCA)[illegible]2[illegible],[illegible],[illegible],[illegible]

[illegible]9.7(a)[illegible]1[illegible]2[illegible]1[illegible],[illegible]2[illegible],

□3□□□□□□□□□□□□,□□□□□□3□□□□,□□□□□□□9.7(b)□,□□□□□□□□□□□

□□□,□□□□□□□□□□□□□□



□9.7　PCA

□□,□□□□□□□□□□□□□□$D_{\text{new}}<D$,□□□□D_{new}□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□$D_{\text{new}}=1$,□□□□□□□□□□□□□□□□□□9.7(c)□□□□□□

□□□□□□□□□,□□□□□□□□□,□□□□□□:□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□2□□3□□□□□□□□□□□□□□□□□□,□□2D□3D□□,□□□□□□□□□□□□□□□□□

9.3.2 UMAP

□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□**t-SNE**□**UMAP**,□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□UMAP□,□□□□□□w□□□:

$$w\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \stackrel{\text{def}}{=} w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)+w_j\left(\boldsymbol{x}_j, \boldsymbol{x}_i\right)-w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) w_j\left(\boldsymbol{x}_j, \boldsymbol{x}_i\right) \tag{9.5}$$

□□$w_i(\boldsymbol{x}_i,\boldsymbol{x}_j)$□□□:

$$w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \stackrel{\text{def}}{=} \exp\left(-\frac{d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)-\rho_i}{\sigma_i}\right)$$

□□,$d(\boldsymbol{x}_i,\boldsymbol{x}_j)$□□□□□□□□□□□□,$\rho_i$□$\boldsymbol{x}_i$□□□□□□□□□,□$\sigma_i$□$\boldsymbol{x}_i$□□□$k$□□□□□□($k$□□□□□□)□

□□9.5□□□□□□□,□□□□□□□□0□1,□□□□□□□□□,$w(\boldsymbol{x}_i,\boldsymbol{x}_j)=w(\boldsymbol{x}_j,\boldsymbol{x}_i)$□

□w□□□□□□□□□□□□□□□□□□;□w'□□□□□□□□□□□□□□□□□□□□,□□9.5 □□□

□□□□□□□,□□□□□□□□□□□(fuzzy set)□□□□□□□□□□□□□□□□□□□□S □□□□□□x□□□□□□□□$\mu_S(x)\in[0,1]$,□□x□□S□□□□□(membership strength)□□□$\mu_S(x)$□□0,□□□□x□□□□□□□□S□□□,□□$\mu_S(x)$□□1,□x□□□ □□□□□S□□□□□$x\in S$□□□$\mu(x)=1$,□□□S□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□w□w'□□□□0□1□□,□□□□□□□$w(\boldsymbol{x}_i,\boldsymbol{x}_j)$□□□□□□$(\boldsymbol{x}_i,\boldsymbol{x}_j)$□□□□□□□ □□□□□□$w'$□□□□□□□□□□□□□□□□□□□□□□□□□(fuzzy set cross-entropy),□□□:

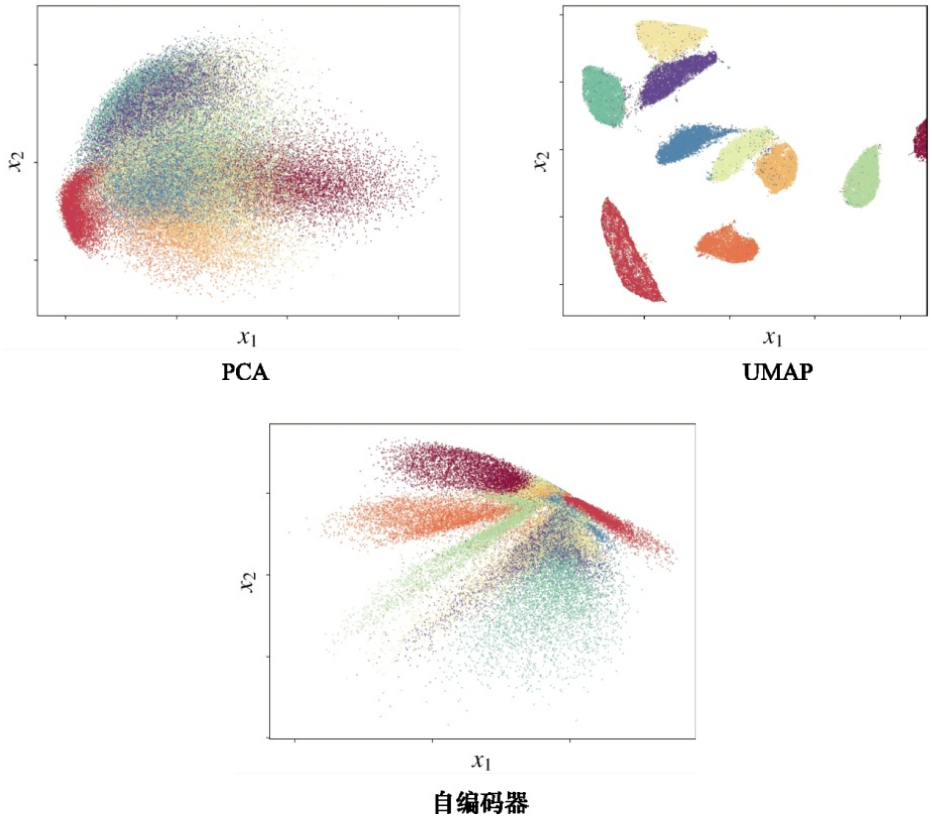
$$C_{w,w'}=\sum_{i=1}^{N}\sum_{j=1}^{N}\left[w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\ln\left(\frac{w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{w'\left(\boldsymbol{x}'_i,\boldsymbol{x}'_j\right)}\right)+\left(1-w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\right)\ln\left(\frac{1-w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{1-w'\left(\boldsymbol{x}'_i,\boldsymbol{x}'_j\right)}\right)\right] \tag{9.6}$$

□□,$\boldsymbol{x}'$□□□□□□□$\boldsymbol{x}$□□□“□□”□

□□9.6□,□□□□□$\boldsymbol{x}'_i$(□□□$i=1,\ldots,N$),□□□□□□□□□□□□□□□□□□□□□□ □□□$C_{w,w'}$□□□

□□9.8□,□□□□□□MNIST□□□□□□□□□□□□□□□□□□MNIST□□□□□□ □□□□□□□□□□:□□□70 000□□□□□□□□9.8□□□10□□□□□□□10□□□□,□□□□□

□□□□□□□□□□□□□□□,UMAP□□□□□□□□□□(□□□□□□□□□□)□□□□
□,UMAP□□PCA□□,□□□□□□□□□



□9.8 □MNIST□□□□□□3□□□□□□□□□□□

9.4 □□□□□

□□□□□(outlier detection)□□:□□□□□□□□□□□□□

□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□

[1] [illegible]

[illegible]10[illegible]

10.1 [illegible]

[illegible],[illegible]([illegible])[illegible](Euclidean distance)[illegible](cosine similarity)[illegible],[illegible]([illegible])[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]k[illegible]kNN[illegible],[illegible]?[illegible],[illegible]

[illegible]$\boldsymbol{x}$[illegible]$\boldsymbol{x}'$[illegible]:

$$d\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left\|\boldsymbol{x} - \boldsymbol{x}'\right\| \stackrel{\text{def}}{=} \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)^2} = \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)\left(\boldsymbol{x} - \boldsymbol{x}'\right)}$$

[illegible],[illegible],[illegible]:

$$d_{\boldsymbol{A}}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_{\boldsymbol{A}} \stackrel{\text{def}}{=} \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{x} - \boldsymbol{x}'\right)}$$

[illegible],$\boldsymbol{A}$[illegible]$D \times D$[illegible]$D=3$,[illegible]$\boldsymbol{A}$[illegible]:

$$\boldsymbol{A} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□d_A□□□□□□□□□□□□□□□□□□□□:

$$\boldsymbol{A} \stackrel{\text{def}}{=} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□□□□□□□□□□□□□(□□□□,□□□□□□□□□□□□)□□□,□□□□□□□□□□□□□□□3□□□:

- $d(\boldsymbol{x},\boldsymbol{x}') \geq 0$ □□□
- $d(\boldsymbol{x},\boldsymbol{x}') \leq d(\boldsymbol{x},\boldsymbol{x}') + d(\boldsymbol{x}',\boldsymbol{z})$ □□□□□
- $d(\boldsymbol{x},\boldsymbol{x}') = d(\boldsymbol{x}',\boldsymbol{x})$ □□□

□□□□□□□□,$\boldsymbol{A}$□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{M}$□□□:

$$\boldsymbol{z}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{z} \geqslant 0$$

□□,$\boldsymbol{z}$□□$\boldsymbol{M}$□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

□□□□□□□□,□□□□□□□□($d(\boldsymbol{x},\boldsymbol{x}')+d(\boldsymbol{x}',\boldsymbol{x}))/2$□

□□□□□□□□□□□□$X = \{\boldsymbol{x}_i\}_{i=1}^{N}$□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□S□□□□□□□□□□□($\boldsymbol{x}_i$,$\boldsymbol{x}_k$);□□□□□D□□□□□□□□($\boldsymbol{x}_i$,$\boldsymbol{x}_k$),□□□□□□□

□□□□□□□□□□$\boldsymbol{A}$,□□□□□□□□□□□□:

$$\min_{\boldsymbol{A}} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in S} \|\boldsymbol{x}-\boldsymbol{x}'\|_{\boldsymbol{A}}^2 \quad \text{□□} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in D} \|\boldsymbol{x}-\boldsymbol{x}'\|_{\boldsymbol{A}} \geqslant c$$

[illegible],***A***[illegible],*c*[illegible]([illegible])[illegible]

[illegible],[illegible],[illegible]***A***[illegible]

[illegible],[illegible](one-shot learning)[illegible](Siamese network)[illegible](triplet loss)[illegible]:[illegible]*S*[illegible];[illegible]*D*[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

10.2 [illegible]

[illegible](learning to rank)[illegible],[illegible],[illegible]*N*[illegible],[illegible]X_i[illegible]r_i[illegible]([illegible])[illegible]*f*,[illegible],[illegible]*f*[illegible]

[illegible]$i=1,\dots,N$,[illegible]X_i[illegible]:$X_i=\{(\boldsymbol{x}_{i,j}, y_{i,j})\}_{j=1}^{r_i}$[illegible]$\boldsymbol{x}_{i,j}$[illegible]$j=1,\dots,r_i$[illegible],$\boldsymbol{x}_{i,j}^{(1)}$[illegible];$\boldsymbol{x}_{i,j}^{(2)}$[illegible];

$x^{(3)}_{i,j}$□□□□□□,□□□□□□□$y_{i,j}$□□□□□$(1,2,\ldots,r_i)$,□□□□□□□□□□,□□□□□□□□□□
□□□□

□□□□□□3□□□:□□□(pointwise)□□□□(pairwise)□□□□□
(listwise)□

□□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□$(\boldsymbol{x},y)$□,$\boldsymbol{x}$□□□□□□□□,y□□□□□(□□$y_{i,j}$□□□□□),□□□□□□□□□□□□□(□□□□,□□□□□□)□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□(□□□□□□□□□□$y_{i,j}$□□)□□□□□□□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□$(\boldsymbol{x}_i,\boldsymbol{x}_k)$,□□□□□□□□$f$□$(\boldsymbol{x}_i,\boldsymbol{x}_k)$□□□□□□□□□□□□$\boldsymbol{x}_i$□□□□□□$\boldsymbol{x}_k$□□,□□□□□□1;□□,□□□□□□0□□□□□,□□□□□□□X□□□□□□□□□X□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□

□□□□□□□□□□□(□**LambdaMART**)□□□□□□□□□□□□□□□,□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(Mean Average Precision,MAP)□

□□□MAP,□□□□□□□□□[□□□□□□□□□□□□(ranker)]□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□(1□□“□□”,0□□“□□”),□□□□□□□,□□□1□5:□□□□□,□□□□□□□□□□□□□□□□□□□□□100□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□:

$$精确度 = \frac{|\{相关文档\} \cap \{搜索返回文档\}|}{|\{搜索返回文档\}|}$$

□□,$|\cdot|$□□□□□□□□□□□□□□□□□□□□□□□□□□q□□□□□□□□□□□□
(Average Precision,AveP)□□□□□□:

$$\mathrm{AveP}(q) = \frac{\sum_{k=1}^{n}(P(k) \cdot \mathrm{rel}(k))}{|\{相关文档\}|}$$

□□,n□□□□□□□;$P(k)$□□□□□,□□□□□□□□□k□□□□□□;rel(k)□□□□□,□k□□□□□□□□□□□(□□□□□□□)□□1,□□□0□□□,Q□□□□□MAP□□□□:

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP}(q)}{Q}$$

□□,□□□□□□LambdaMART□□□□□□□□□□□,□□□□□□□□□□□$h(\boldsymbol{x})$□□□,□□□□□□$f(\boldsymbol{x}_i,\boldsymbol{x}_k)$□□□□$\boldsymbol{x}_i$□□□$\boldsymbol{x}_k$□□□□(□□□□□□□□)□□□□□□□□□□□□α□sigmoid□□□

$$f(\boldsymbol{x}_i,\boldsymbol{x}_k) \stackrel{\mathrm{def}}{=} \frac{1}{1+\exp(h(\boldsymbol{x}_i)-h(\boldsymbol{x}_k))\alpha}$$

□□,□□□□□□□□□□□□□,□□□□□□□□□f□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□h□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□LambdaMART□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□MAP□□□□□□□□□□□□□□□□□□□

[illegible],[illegible],[illegible]
[illegible],[illegible]([illegible])[illegible],[illegible]
[illegible],[illegible]LambdaMART[illegible]

[illegible],[illegible]f[illegible]f[illegible]
[illegible],[illegible],[illegible],[illegible]

[illegible](sorting)[illegible]
[[illegible](bubblesort)[illegible]][illegible],[illegible]
[illegible],[illegible]f[illegible],
[illegible]

10.3 [illegible]

[illegible],[illegible]
[illegible],[illegible],[illegible](Netflix)[illegible],[illegible](Amazon)
[illegible]

[illegible]:[illegible](content-based filtering)[illegible](collaborative filtering)[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible] ***x***,[illegible] *y* [illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible] 10 [illegible],[illegible] 1 [illegible] 2 [illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible](0)[illegible],[illegible]

[illegible]:[illegible]

[illegible],[illegible]([illegible])[illegible]

[illegible]:[illegible](Factorization Machines,FM)[illegible](Denoising Autoencoders,DAE)[illegible]

10.3.1 [illegible]

[illegible],[illegible]

[illegible]10.1[illegible],[illegible],[illegible]x_{99}[illegible]x_{100}[illegible]y[illegible]

	用户				电影					电影评价								
	Ed	Al	Zak	···	It	Up	Jaws	Her		It	Up	Jaws	Her					
	x_1	x_2	x_3	···	x_{21}	x_{22}	x_{23}	x_{24}	···	x_{40}	x_{41}	x_{42}	x_{43}	···	x_{99}	x_{100}	y	
$\boldsymbol{x}^{(1)}$	1	0	0	···	1	0	0	0	···	0.2	0.8	0.4	0	···	0.3	0.8	1	$y^{(1)}$
$\boldsymbol{x}^{(2)}$	1	0	0	···	0	1	0	0	···	0.2	0.8	0.4	0	···	0.3	0.8	3	$y^{(2)}$
$\boldsymbol{x}^{(3)}$	1	0	0	···	0	0	1	0	···	0.2	0.8	0.4	0	···	0.3	0.8	2	$y^{(3)}$
$\boldsymbol{x}^{(4)}$	0	1	0	···	0	0	1	0	···	0	0	0.7	0.1	···	0.35	0.78	3	$y^{(4)}$
$\boldsymbol{x}^{(5)}$	0	1	0	···	0	0	0	1	···	0	0	0.7	0.1	···	0.35	0.78	1	$y^{(5)}$
$\boldsymbol{x}^{(6)}$	0	0	1	···	1	0	0	0	···	0.8	0	0	0.6	···	0.5	0.77	4	$y^{(6)}$
⋮	⋮	⋮	⋮												⋮	⋮	⋮	⋮
$\boldsymbol{x}^{(D)}$	0	0	0	···	0	0	1	0	···	0	0	1	0	···	0.95	0.85	5	$y^{(D)}$

[illegible]10.1 [illegible]$\boldsymbol{x}$[illegible]y

[illegible],[illegible],[illegible]10.1[illegible],[illegible],[illegible]x_{99}[illegible]x_{100}[illegible],[illegible]

[illegible],[illegible]

数学模型的公式如下所示:

$$f(\boldsymbol{x}) \overset{\text{def}}{=} b + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=i+1}^{D} (\boldsymbol{v}_i \boldsymbol{v}_j) x_i x_j$$

其中,b和w_i(i=1,…,D)与线性相似,我们引入了附加参数。向量$\boldsymbol{v}_i$是一个包含k个因子(factor)维度的k维的向量,属于超D参数。$\boldsymbol{v}_i\boldsymbol{v}_j$是第$i$个和第$j$个特征之间的关系。

我们,将参数矩阵低秩分解为两个,而不是像这样,拟合一整个成对交互参数矩阵。理论上我们可以对每个交互特征x_ix_j使用单独的参数进行建模,也就是一个交互作用的参数$w_{i,j}$,但是这样有很大的问题[1]。我们的方法通过分解参数,能减少参数的数量到Dk<<D(D-1)的数量[2],我们用$w_{i,j}$来表示$\boldsymbol{v}_i\boldsymbol{v}_j$。

在具体问题上,根据目标的不同使用二分类(回归)损失函数。对于标签y∈{-1,+1}的二分类问题,我们使用逻辑损失函数来进行分类,预测y=sign(f(x))。逻辑损失函数如下所示:

$$\text{loss}(f(\boldsymbol{x}), y) = \frac{1}{\ln 2} \ln \left(1 + \mathrm{e}^{-yf(\boldsymbol{x})}\right)$$

对于多类别的分类问题来说,比如表10.1中的评分,它的取值{1,2,3,4,5},我们可以把它转化为二分类的一对其余(one versus rest)问题,这样就可以训练5个二分类模型。

10.3.2　问题的简化方法

第7章已经讲过一种方法:对于多分类的问题,我们可以将其转化为多个二分类问题的集合,然后对每个问题训练一个二分类模型,最后用模型预测的结果来选择。

另一种方法:我们可以在预测类别的时候使用回归模型来预测评分大小然后进行四舍五入,这可以用于有序类别,比如评分预测类的问题。

□□□□□□□□□□□□,□□□10.1□□□□□□□□□□□□□□□□□□□□,□
□□□□□□□□□

□□□□,□□□□□□□,□□□□□□0□□□□□0□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□x_{99}□x_{100}□□□
□□□*DAE*□□□□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□*FFNN*□□8□□□,□□□□□□□□
□□□□□□□□□,□□□□□□□□□□□(***u***,***m***,*r*)□□□□***u***□□□□□□□□□□□□□
□***m***□□□□□□□□□□*r*□□□□[0,1]□,□□□□□sigmoid;□□□□□□□□□
□,□[1,5]□,□□□□ReLu□

10.4 □□□□□:□□□

□7□□□□□□□□□□(word embedding)□□□□□□□□□□□□,□□□
□□□□□□□□□□□□□□,□□□□□□□□□?□□(□□)□:□□□□□□

□□□□□□□□□□□□□□,□□□□□□□□□word2vec,□□word2vec□□
□□□□□□□□□□□□□(skip-gram)□□□□□□□□□□□□□□□word2vec

□□□

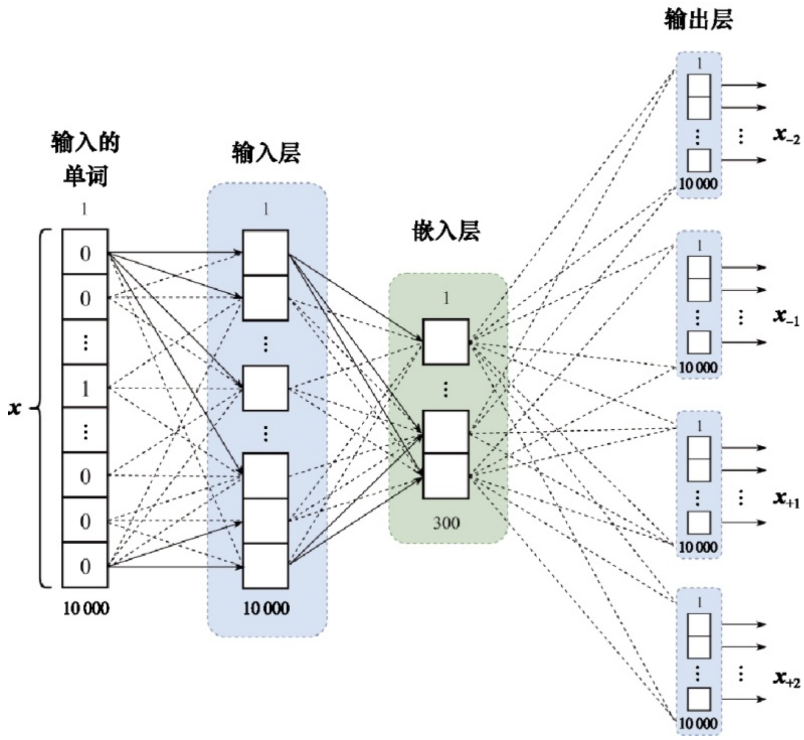
□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□10 000□□□□□□□□□□□□□□10 000□□□,□□1□□□□□□□1□□,□□□□□□□□□0□

□□□□□□“I almost finished reading the book on machine learning.”(□□□□□□□□□□□□□□)□□[3],□□□□□□,□□“book”(□),□□□□□□,□□□□□□□□“I almost finished reading the • on machine learning.”□□□,□□□□□•□□□□□□□3□□□“finished reading the ? on machine learning”□□□□□□□□7□□□□□□,□□□□“?”□□□□,□□□□□□“book”(□)“article”(□□)□“paper”(□□)□□□□□□□□□□□□□□□□□□□□“book”“article”□“paper”□□□□□□□□□:□□□□□□□□□□□□□□

□□□,□□□□□□□□□□□□□□□□□□“finished reading the•on machine learning”□□□□□□□□□7(3+1+3)□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□[$\boldsymbol{x}_{-3}$,$\boldsymbol{x}_{-2}$,$\boldsymbol{x}_{-1}$,$\boldsymbol{x}$,$\boldsymbol{x}_{+1}$,$\boldsymbol{x}_{+2}$,$\boldsymbol{x}_{+3}$]□□□□□□□□□□□□,$\boldsymbol{x}_{-3}$□□□“finished”(□□)□□□□□,$\boldsymbol{x}_{-2}$□□“reading”(□),$\boldsymbol{x}$□□□•,$\boldsymbol{x}_{+1}$□“on”,□□□□□□□□□□□□□5□□□□□□□□□□□:[$\boldsymbol{x}_{-2}$,$\boldsymbol{x}_{-1}$,$\boldsymbol{x}$,$\boldsymbol{x}_{+1}$,$\boldsymbol{x}_{+2}$]□

□10.2□□□□□□□□□□5□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□•□□□□□□□□□□□□□□□□□□□□□



□10.2　□□□□□□□5□□□□□□□□300□□□□□

□□□□□□□□□□□□□□□□□□□□(self-supervised):□□□□□□□□□□□□□□□□□□,□□□□

□□□□□□□□□□□softmax□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

word2vec□□□□□□□□,□□□□□□□□□□□□:□□softmax(hierarchicalsoftmax)(□□□□softmax□□□□□,□softmax□□□□□□□□□□□□□□□)□□□□(negativesampling)(□□□□□□□□□□□,□□□□□□□□□□□□□□□□)□□□□□□□□□□□□□□□

---

[1] □□□□□,□□□□□$D(D\text{-}1)$□□□$w_{i,j}$□

[2] <<□□□□□□□□□□□

[3] □□□:□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□,□□□□□□□□,□□□□□□□

[illegible]11[illegible] [illegible]

[illegible],[illegible]![illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible],[illegible]

[illegible],[illegible]?

11.1 [illegible]

[illegible],[illegible]:[illegible],[illegible](Latent Dirichlet Allocation,LDA)[illegible],[illegible],[illegible],[illegible],[illegible]

11.2 [illegible]

[illegible](Gaussian Process,GP)[illegible],[illegible],[illegible],[illegible],[illegible]

11.3 广义线性模型

广义线性模型(Generalized Linear Model,GLM)是逻辑回归的推广,它允许因变量服从指数族中的任意一种概率分布。这本书中的大部分内容都与之相关,理解什么情况下可以使用GLM并不是一件太有挑战性的事,但更全面地理解它的数学基础,可以帮助你更好地运用GLM。

11.4 概率图模型

此外,第7章还介绍了一种概率图模型(Probabilistic Graphical Model,PGM):条件随机场(Conditional Random Field,CRF)。通过CRF可以了解到依赖关系图(dependencygraph)的概念。依赖关系图是一种描述变量之间依赖关系的一种优雅的方式,这是PGM的核心理念。使用图(graph)描述变量之间的关系是一种很直观的方式。图中节点对应随机变量(有些变量可能是隐藏的),边表示变量之间的依赖关系。图可以帮助我们理解模型的结构,理解哪些“变量条件独立”以及如何基于“消息传递”的算法去推断变量的分布,因而图模型为概率建模提供了一种统一、优雅的框架。

PGM是一个有着坚实理论基础的研究领域。很多模型都可以用概率图模型的形式表示,比如朴素贝叶斯就可以看作一种特殊的图模型,因而掌握概率图模型可以加深对各种机器学习模型的理解,进而可以在实际问题中灵活地构建自己的模型。概率图模型的学习和推断算法非常丰富,其中的一些算法比较复杂,并且需要很多数学知识。比如你会发现,虽然我们在本书中介绍的CRF只是一种很简单的PGM,但是训练的过程依然不是那么简单,在实际应用中,CRF通常还需要配合其他模型使用。比如一种常用的隐马尔可夫模型(Hidden Markov Model,HMM),它也是处理序列数据的一种经典方法。在很多序列标注问题中,HMM都有着非常广泛的应用。

□□□□□□□□□PGM,□□□□□(Bayesion network)□□□□□(belief network)□□□□□□□(probabilistic independence network)□□□□□□□□

11.5 □□□□□□□□□□

□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□(Markov Chain Monte Carlo,MCMC)□□□MCMC□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□(denoising autoencoder)□,□□□□□□□□□□□□□□□□□(□□□□□□□□□□□)□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

11.6 □□□□

□□□□(Genetic Algorithm,GA)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□)□

□□□,GA□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□“□□”“□□”□“□□”□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□“□□”□“□□”□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□(□□□□)□

[illegible],GA[illegible],[illegible]

11.7 [illegible]

[illegible],[illegible](Reinforcement Learning,RL)[illegible],[illegible],[illegible](agent)[illegible](reward),[illegible]([illegible])[illegible]

[illegible],[illegible]Q[illegible](Q-learning),[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]

A

accuracy([illegible]) [illegible],[illegible]

activation function([illegible]) [illegible],[illegible],[illegible]

active learning([illegible]) [illegible],[illegible],[illegible],[illegible]

AdaBoost[illegible]AdaBoost[illegible],[illegible],[illegible],[illegible]

agglomerative clustering algorithm([illegible]) [illegible]“[illegible]”[illegible],[illegible]

area under the ROC curve(**ROC**[illegible]) [illegible]ROC[illegible]([illegible],[illegible])[illegible]([illegible]),[illegible]ROC[illegible],[illegible],[illegible]

autoencoder([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]。

B

backpropagation([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]。

bag of words([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]。

bagging([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]。

base learner([illegible]) [illegible],[illegible],[illegible]。

baseline([illegible]) [illegible](heuristic)[illegible]。

batch([illegible]) [illegible](minibatch),[illegible]。

batch normalization([illegible]) [illegible],[illegible]0[illegible]([illegible]:[illegible])[illegible],[illegible],[illegible],[illegible](regularization)[illegible]

Bayes’ Rule([illegible]) [illegible],[illegible]18[illegible]•[illegible],[illegible]

Bayesian optimization([illegible]) [illegible],[illegible]

bias([illegible]) [illegible]([illegible])[illegible]

bias-variance tradeoff([illegible]**-**[illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible]-[illegible]

bigram([illegible]([illegible])[illegible]) [illegible],[illegible]“cat”[illegible],[illegible]“ca”[illegible]“at”[illegible]“[illegible]”[illegible]3[illegible]:“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible],[illegible],[illegible]

bin([illegible]) [illegible],[illegible]1[illegible]

binary classification([illegible]) [illegible],[illegible],[illegible]

binary variable(二元变量)　只有两个可能取值的类型变量(或分类变量):1或0,“是”或“否”。

binning(分箱)　也称分桶,将一组数值数据转换成一组类型变量(或分类变量),常常为了让数据更有可比性,或者为了在线性模型中引入非线性,而把数值型的数据划分为若干类型。例如:以下拥有0~5次交易的客户为一组,6~10次交易的为一组,11~15次交易的为一组,16~25次交易的为一组,以此类推。

boosting(提升)　是一种迭代处理过程,它通过在一系列实验中逐渐增加错误记录(训练样本)的权重或者误分类记录(训练样本)的权重,从而使得后续模型特别关注那些训练过程中出错或误分类的记录,最有名的是AdaBoost等提升算法。

bootstrap aggregating(自举法聚集)　见Bagging(装袋)。

Bootstrapping(自助法)　以自助采样法为基础的处理方法,给定包含某个数量样本的数据集,对它进行有放回地采样,每次随机从数据集中挑选一个样本放入训练数据集中,然后再将该样本放回初始数据集中,使得该样本在下次采样时仍可能被采到,从而产生所需要的训练数据集。

Bottleneck(瓶颈)　是指一个系统中的组成部分,其限制着整个系统的性能或者运行能力,在计算机系统中通常指的是那些限制整个处理过程的组件,例如存储器、处理器或者网络带宽。

bucket(桶)　见bin(箱)。

bucketing(分桶)　见binning(分箱)。

C

C4.5　是一种流行的决策树算法,类似于ID3,但有如下改进:可以处理连续值属性和带缺失值的数据集,还可以对生成的决策树“剪枝”以避免过拟合等问题。

CART(□□□□□□) □□□□□□□□□□□□,□C4.5□□□

categorical feature(□□□□) □□□□□□□□□□□□□□

categorical variable(□□□□) □□□□□□□□□□□,□□□□□□□□□□□□□,□□□□“□□□□□”□□3□□□□:“□”“□”□“□”□

centroid(□□) k□□□k□□□□□□□□□□□□□□□□□□□□,□□*k*□□4,k□□□k□□□□□□4□□□□

class(□□) □□,□□□□□□□□□□□□□□,□□□□□□□“□□□□”□“□□□□□”□□□□□□□□□□□□□□,□□□□□“□”“□”“□□”□□

classification(□□) □□□□□□□□□□□□□

classification algorithm(□□□□) □□□□□□□□,□□□□□□□□□□□□□□□□□□□□

classification model(□□□□) □classification(□□)□

classification threshold(□□□□) □□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□0.9,□□□□□□0.9□□□□□□,□□0.9□□□□□□

classifier(□□□) □model,classification(□□,□□)□

Cluster(□□□) □□□□□□□□□□□□□□□□(□□□□□□□□□□□□□)□

Clustering(□□)　□□□□□□□□□□□□□□□□

clustering algorithm(□□□□)　□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□k□□□□□□□□DBSCAN□□

computational graph(□□□)　□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

confusion matrix(□□□□)　□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□“□”□□□□□“□”,□□“□”□□□“□”□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□

continuous variable(□□□□)　□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

convergence(□□)　□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□

convolution(□□)　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

convolution filter([illegible]) [illegible],[illegible]

[illegible]

convolutional neural network(**CNN**,[illegible]) [illegible]

[illegible],[illegible],CNN[illegible]

[illegible],[illegible],[illegible]

[illegible]

cosine similarity([illegible]) [illegible],[illegible]

[illegible]

cost function([illegible]) [illegible],[illegible],[illegible]

([illegible])[illegible]

cross entropy([illegible]) [illegible]X[illegible]p

[illegible]q,[illegible]$H(p,q)=-\sum_{x\in X}p(x)\log q(x)$[illegible],[illegible]

[illegible]

cross-validation([illegible]) [illegible]

[illegible],[illegible],[illegible]

[illegible]

D

dataset([illegible]) [illegible]

DBSCAN [illegible],[illegible]

decision boundary([illegible]) [illegible],[illegible]
[illegible],[illegible]

decision tree([illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible]ID3[illegible]C4.5[illegible]CART[illegible]

decision tree learning algorithm([illegible]) [illegible]ID3[illegible]CART[illegible]C4.5[illegible]

deep neural network([illegible]) [illegible]

density-based clustering algorithm([illegible]) [illegible],[illegible]([illegible]),[illegible],[illegible]

dimensionality reduction([illegible]) [illegible],[illegible]([illegible])[illegible](Principal Component Analysis,PCA)[illegible]UMAP[illegible],[illegible]word2vec[illegible],[illegible],[illegible],[illegible]

divisive clustering algorithm([illegible]) [illegible]:[illegible],[illegible],[illegible]

dropout([illegible]) [illegible],[illegible]0,[illegible]

dummy variable([illegible]) [illegible]0[illegible]1[illegible],0[illegible]25,1[illegible]25[illegible]:binning([illegible])[illegible]

E

early stopping([illegible]) [illegible],[illegible],[illegible]

embedding([illegible]) [illegible],[illegible],[illegible]([illegible]),[illegible]word2vec[illegible]GloVe[illegible],[illegible],[illegible],[illegible],[illegible]

ensemble learning([illegible]) [illegible]

ensemble learning algorithm([illegible]) [illegible]([illegible])[illegible]

epoch([illegible]) [illegible]

evaluation metric([illegible]) [illegible],[illegible]ROC[illegible]F[illegible]

evolutionary algorithm([illegible]) [illegible],[illegible]“[illegible]”“[illegible]”[illegible]

□□

evolutionary optimization(□□□□)　□□□□□□□□□□□□□□□

□,□□□□□□□□□□□□□□□

example(□□)　□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□(□□),□□□□□□□

□□□(□□),□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□

□□

example,**labeled**(□□,□□□)　□□□□□□□□□□□□,□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□,labeled example(□□□□□)□

example,**training**(□□,□□)　□□□□□□□□□

example,**unlabeled**(□□,□□□)　□□□□□□□□□,□□□□□□□□□

□,unlabelled example(□□□□□)□

exploding gradient problem(□□□□□□)　□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□

F

F score(**F**□□)　□F1 score(F1□□)□

F1 score(**F1**□□)　□□F□□F□□,□□□□□□□□□□□□□□□,□□□□□□□

□□□

false positive(**FP**,[illegible]) [illegible],[illegible]([illegible]),[illegible]

feature([illegible]) [illegible],[illegible],[illegible],[illegible]:[illegible]([illegible]),[illegible]([illegible]),[illegible]([illegible])[illegible]

feature engineering([illegible]) [illegible]([illegible])[illegible]:[illegible],[illegible]

feature selection([illegible]) [illegible],[illegible]

feature vector([illegible]) [illegible],[illegible]

feature,**informative**([illegible],[illegible]) [illegible]

feedforward neural network(**FFNN**,[illegible]) [illegible],[illegible],[illegible]([illegible]),[illegible]

few-shot learning([illegible]) [illegible],[illegible]

G

gate([illegible]) [illegible],[illegible]

gated recurrent unit(**GRU**,[illegible]) [illegible],[illegible],GRU[illegible],[illegible]

[illegible]GRU[illegible]([illegible])[illegible]([illegible])[illegible]

gated unit([illegible])　[illegible],[illegible]“[illegible]”[illegible](GRU)[illegible](LSTM)[illegible]

Gaussian distribution([illegible])　[illegible],[illegible]$f_{\mu,\sigma^2}(x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$,[illegible]$\mu$[illegible]([illegible]),σ^2[illegible]

Gaussian process([illegible])　[illegible],[illegible]

generalization([illegible])　[illegible]

genetic algorithm(**GA**,[illegible])　[illegible],[illegible],[illegible]([illegible])[illegible],GA[illegible],[illegible],[illegible],[illegible]“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible]*d*[illegible]

GloVe([illegible])　[illegible],[illegible]word2vec[illegible]

gradient boosting([illegible])　[illegible],[illegible]

gradient clipping([illegible]) [illegible],[illegible],[illegible]

gradient descent([illegible]) [illegible],[illegible],[illegible]:[illegible](SGD)[illegible]

gradient step([illegible]) [illegible],[illegible]

grid search([illegible]) [illegible],[illegible]

H

hidden layer([illegible]) [illegible],[illegible]

hidden unit([illegible]) [illegible],[illegible]

hierarchical clustering algorithm([illegible]) [illegible]:[illegible]:[illegible]

hinge loss([illegible]) [illegible],[illegible]

holdout set([illegible]) [illegible],[illegible](“[illegible]”)[illegible],[illegible]

hyperparameter([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]:[illegible]

hyperparameter tuning([illegible]) [illegible],[illegible],[illegible]

Hyperplane([illegible]) [illegible],[illegible];[illegible],[illegible],[illegible],[illegible]

I

ID3[illegible]

input layer([illegible]) [illegible],[illegible]

instance([illegible]) [illegible]

instance-based learning([illegible]) [illegible],[illegible],[illegible],[illegible]k[illegible]

Iteration([illegible]) [illegible],[illegible],[illegible]

K

k Nearest Neighbor(**kNN**)(**k**[illegible]) [illegible],[illegible],[illegible]k[illegible],[illegible]k[illegible]

k-mean(**k-**[illegible]) [illegible],[illegible],[illegible]k[illegible]

k-median(**k-**[illegible]) [illegible],[illegible],[illegible]k[illegible]

Kernel([illegible]) [illegible]x'[illegible]x_j[illegible]

kernel trick([illegible]) [illegible]x[illegible]x'[illegible],[illegible]$\varphi(x)$[illegible]$\varphi(x')$[illegible]

L

L1 regularization(**L1**[illegible]) [illegible],[illegible]

L2 regularization(**L2**[illegible]) [illegible],[illegible]

label([illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible]

labeled example([illegible]) [illegible]example,labeled([illegible],[illegible])[illegible]

Latent Dirichlet Allocation(**LDA**,[illegible]) [illegible],[illegible]

latent semantic Indexing([illegible]) [illegible],[illegible]

layer([illegible]) [illegible],[illegible]

learning algorithm([illegible]) [illegible],[illegible]

learning rate([illegible]) [illegible],[illegible],[illegible]

linear regression([illegible]) [illegible],[illegible]

log loss([illegible]) [illegible]

logistic regression([illegible]) [illegible],[illegible],[illegible],[illegible]sigmoid[illegible]

“[illegible]”[illegible],[illegible]“[illegible]”[illegible]

long short-term memory(**LSTM**)**unit**([illegible]) [illegible],[illegible],LSTM[illegible]LSTM[illegible]3[illegible](“[illegible]”“[illegible]”[illegible]“[illegible]”)[illegible]RNN[illegible],[illegible]LSTM[illegible]LSTM[illegible],[illegible]RNN[illegible]RNN[illegible]

loss([illegible]) [illegible]

loss function([illegible]) [illegible],[illegible]:[illegible],[illegible];[illegible]

LSTM network([illegible]) [illegible]LSTM[illegible]

M

machine learning([illegible]) [illegible],[illegible],[illegible],[illegible]:[illegible],[illegible]

machine learning algorithm([illegible]) [illegible]machine learning([illegible])[illegible]

maximum margin classification([illegible]) [illegible],[illegible],[illegible]

metric learning([illegible]) [illegible],[illegible],[illegible]“[illegible]”:[illegible],[illegible],[illegible]

minibatch([illegible]) [illegible]batch([illegible])[illegible]

minibatch stochastic gradient descent([illegible]) [illegible],[illegible],[illegible]“[illegible]”[illegible]

model([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

model,classification([illegible],[illegible]) [illegible],[illegible]

model,regression([illegible],[illegible]) [illegible],[illegible]

multi-class classification([illegible]) [illegible],[illegible]“[illegible]”“[illegible]”“[illegible]”[illegible],[illegible]([illegible]“[illegible]”[illegible]“[illegible]”)[illegible]

multi-class logistic regression([illegible]) [illegible],[illegible]

N

neural network([illegible]) [illegible]([illegible])[illegible]

neuron([illegible]) [illegible]unit([illegible])[illegible]

normalization([illegible]) [illegible],[illegible][-1,+1][illegible][0,1][illegible]

O

objective function([illegible]) [illegible],[illegible]

one-hot encoding([illegible]) [illegible],[illegible]1,[illegible]0[illegible]

outlier([illegible]) [illegible]

output layer([illegible]) [illegible],[illegible]

output unit([illegible]) [illegible]

overfitting([illegible]) [illegible],[illegible],[illegible]

P

parameter([illegible]) [illegible],[illegible],[illegible],[illegible]

partitional clustering algorithm([illegible]) [illegible],[illegible],[illegible]k[illegible]k[illegible]

pooling([illegible]) [illegible]

population([illegible]) [illegible],[illegible]([illegible])[illegible],[illegible],[illegible]

precision(□□□)　□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□:recall(□□□)□confusion matrix(□□□□)□

predictive model(□□□□)　□□□□□□□□□□□□□□□□□

predictive power(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□

principal component analysis(**PCA**,□□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□PCA□□□□□□□□□,□□□□□□□□□□□□□□□□□

R

random forests(□□□□)　□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□

random search(□□□□)　□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

random variable(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□

recall(□□□)　□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□:precision(□□□)□confusion matrix(□□□□)□

recurrent neural network(**RNN**,□□□□□)　□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□RNN□□□□□□□,□□□□□□□□□□□

regression(□□)　□□□□□□□□,□□□□□□□□□□(□□)□□□□

regression algorithm(□□□□)　□□□□□□□□□□□□□□□

regression model(□□□□)　□model,regression(□□,□□)□

regularization(□□□)　□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□L1□□□□L2□□□□□□□□

reinforcement learning(□□□□)　□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□

representation learning(□□□□)　□□□□□□□,□□□□□□□□□□□□□

risk(□□)　□□□□□□□□□

S

semi-supervised learning(□□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□

sigmoid function(**S**□□□)　□□□□□□□□□□□□□□(0,1)□□□□□□□□□□□□□□□□□□□

similarity metric(□□□□□)　□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□,□□□□□□□□□

standardization(□□□)　□□z□□□,□□□□□□□□□□□□□□□□μ=0□σ=1□□□□□□□□□□□□□□□,μ□□□(□□□□□□□□□□□□□□□□),σ□□□□□

statistic(□□)　□□□□□□□□□,□□□□□□□□

statistical model(□□□□)　□model(□□)□

stochastic gradient descent(**SGD**,□□□□□□)　□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□SGD□□□□□

strong classifier(□□□□)　□□□

supervised learning(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□

Support Vector Machine(**SVM**,□□□□□)　□□□□□□,□□□□□□□□□□□□□□□□□□SVM□□□□□□□□□□□□□

T

target(□□)　□□□□□□□□□□□□□□□□□□□□

test set(□□□)　□□□□□□□□□□□□□□□□□:training set(□□□)□validation set(□□□)□

token(□□)　□□□□□□□□□□□,□□□□□□□□

tokenization(□□□)　□□□□□□□□□□,□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□

topic modelling(□□□□)　□□□□□□□□□□□□□□□,□□□□□□□□□
□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
(LDA)□□□□□□(LSI)□

training(□□)　□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□

training data(□□□□)　□traing set(□□□)□

training example(□□□□)　□example,training(□□,□□)□

training loss(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□

training set(□□□)　□□□□□,□□□□□□□□□□,□□□□□□□□□□□
□:test set(□□□)□validation set(□□□)□

transfer learning(□□□□)　□□□□□□□,□□□□□□□□□□□□□□□□
□□□□□□□

true negative(**TN**,□□)　□□□□□□□,□□□□□□□□□□□□□□□□□□

true positive(**TP**,□□)　□□□□□□□,□□□□□□□□□□□□□□□□□□

U

UMAP(□□□□□□□□) □□□□□□□,□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□

underfitting(□□□) □□□□□□□□□□□□□,□□□□□□□□□□□□
□□□□

unit(□□) □□□□□□□□,□□□□□□□□,□□□□□□□□,□□□□□□□
□□□□□□□□□□□□□(□□□□)□□□□□□□□□□□□□□

unlabeled example(□□□□) □example,unlabeled(□□,□
□□)□

unsupervised learning(□□□□) □□□□□□□□□□□□□(□
□□)□□□□□□□□□□□□□□□□□□□□□□□□

unsupervised learning algorithm(□□□□□□) □□□□□
□□□□□□□□

V

validation example(□□□□) □□□□□□

validation loss(□□□□) □□□□□□□□□□□□□□□□□□□□□

validation set(□□□) □□□□□□□□□□□

vanishing gradient problem(□□□□□) □□□□□□□□□□
□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□

[illegible]“[illegible]”,[illegible]

variance([illegible]) [illegible]
[illegible],[illegible]([illegible])[illegible]

volume([illegible]) [illegible],[illegible]
[illegible],[illegible](CNN)[illegible]

W

weak classifier([illegible]) [illegible],[illegible]
[illegible],[illegible]

word embedding([illegible]) [illegible];[illegible]
[illegible],[illegible],[illegible]

word2vec [illegible],[illegible],[illegible],[illegible]
[illegible]:GloVe([illegible])[illegible]

Z

zero shot learning([illegible]) [illegible],[illegible]
[illegible],[illegible]